TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JAKUB EŠNER

DEVELOPMENT OF "AC MICROGRID" LAB ENVIRONMENT - CASE STUDY OF PRODUCTION FOLLOWING CONSUMPTION

Master of Science Thesis

Examiner: Professor Sami Repo
Examiner and topic approved by the Council of the Faculty of Computing and Electrical Engineering on 9.4.2014

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Electrical Engineering
EŠNER, JAKUB: Development of "AC Microgrid" Lab Environment – Case Study of Production Following Consumption
Master of Science Thesis, 64 pages, 4 Appendix pages
February 2015
Major: Smart Grids
Examiner: Professor Sami Repo
Keywords: Production following, Demand response, Load shifting, HIL simulations, Power measurement, prototype

Photovoltaic installations on resident's rooftops are becoming increasingly popular. An increased deployment of intermittent power generators requires new installations of controllable resources in order to minimize stress introduced to electrical grid. The best solution is creating these resources in houses with photovoltaic installation. A production following consumption can be implemented e.g. by controlling space heaters or energy storage systems.

In this thesis, a production following algorithm is developed utilizing heat energy storage. Load shifting implementation requires the same equipment and therefore has been combined with production following algorithm. The algorithm first generates schedules based on day-ahead-prices and weather forecast. These schedules are then interpreted by real-time controllers directly interfacing with appliances in order to achieve production following while maintaining pre-set room temperatures.

In order to evaluate algorithm's performance a mathematical model of household had to be created. This model comprises of thermal model of house, hot water boiler and radiator. The case study simulations showed that developed algorithm utilizing hot water boiler as energy storage can offer considerable savings. A return of investments can be made in less than 7 years with total savings around 840€. Moreover, no support from network operators has been considered and thus higher revenue could be achieved.

Simulated results might be doubted as the mathematical model of house and its appliances cannot mimic all aspects of real-life environment. For that reason, a laboratory has been build. The laboratory should be able to emulate behavior of real-word house so different algorithm implementations could be objectively tested in realistic environment. The core of the laboratory is AC Microgrid panel equipped with seven universal three-phase channels. Each of these channels can be controlled by dSpace, which is responsible for environment emulation, or by home-energy-management computer utilizing wireless Z-Wave interface. A measurement system has been designed so power-flows within household could be provided to home-energy-management computer and its algorithms. A Matlab / Simulink interface blocks have been implemented so an easy transition from simulations to hardware-in-the-loop simulations is possible.

PREFACE

This Master of Science thesis was done for the Department of Electrical Engineering at Tampere University of Technology. The supervisors and examiner of the thesis have been Prof. Sami Repo and Prof. Pertti Järventausta.

I wish to thank Sami and Pertti for providing me this interesting topic. Great thanks also to all my colleagues of the Department, they have made me feel like at home despite the language and cultural differences. Finally, I want to express my sincere gratitude for my family for the support during my studies and this work.

Tampere 04.05.2015

Jakub Ešner

TABLE OF CONTENTS

TERMS AND DEFINITIONS

API	Application programming interface
ARM	Advanced RISC machine
BMS	Battery management system
CAN	Controller area network
CSV	Comma separated values
CT	Current transformer
DAP	Day ahead pricing
DER	Distributed energy resource
DG	Distributed generation
DIED	Distributed Intelligent Electronic Device
DR	Demand response
DSO	Distribution system operator
ESS	Energy storage system
EV	Electric vehicle
GND	Ground
HEM	Home energy management
HEMS	Home energy management system
HIL	Hardware in the loop
LFP	Lithium ion phosphate
LMTD	Logarithmic mean temperature difference
N/C	Not connected
OZW	Open Z-Wave
PC	Personal computer
PLC	Power line cycle
PV	Photovoltaic
PWM	Pulse width modulation
RAM	Random access memory
RISC	Reduced instruction set computing
RTDS	Real Time Digital Simulator
SAU	Substation Automation Unit
SGEM	Smart Grids and Energy Markets
SoC	System on the chip
SSR	Solid state relay
TOU	Time of use
TTL	Transistor-transistor logic
TVS	Transient voltage suppressor

1. INTRODUCTION

Today's society is dependent on energy more than ever before. This strong dependency on stable electricity delivery is one of the biggest driving forces behind re-design of whole power system. The technology that is now accessible to us made design of power system that dynamically responses to its state, in order to optimize its stability and efficiency, possible. Applying this new approach on electricity market and network design gave definition to smart grid.

Stability of power supply can be increased by utilization of distributed generation (DG) placed close to customer. The short distribution path lowers likelihood of its failure and thus the increase in stability. Another way of increasing power supply stability can be done by avoiding generation that demands fuel import. This kind of dependency was felt in January 2009 when a disagreement between Ukraine and Russia led to disruption in natural gas supply for many European countries. The energy dependence of Finland is quite high 45% [1]. Energy of 132 000TJ was imported only in natural gas in 2013 [1]. To produce this amount of energy a continuous production of 4.2GW is required.

Moreover approximately 4 billion metric tons of CO_2 is produced annually by electric power generation. It has been predicted that the rise of global temperature will be between 2 and 5 degrees Celsius in next 100 years. According to the worst-case scenarios melting glaciers might result in sea level rising by one meter within 100 years. This is alarming observation, since about 100 million people would have to move inland. Global warming will also cause damage to vegetation and agriculture because of droughts. Extreme weather conditions will happen more often and gulfstream might change substantially causing freezing weather in some parts of the world. [2] Therefore, an extensive installation of non-polluting renewable distribution generation is sensible precaution.

A photovoltaic array installation on rooftops of households is becoming increasingly popular for good reasons. It is placed at the customer's place, so distribution losses are minimal, the installation is simple as many companies are providing system as turn-key solution and last but not least it is becoming profitable even without financial support.

Another positive aspect of DG placed close to consumption is that it can solve line's congestion problem and therefore defer investments for network's extension. However, in order to ensure that the intermittent DG units like solar or wind will not worsen the congestion a small-scale reserve has to be installed [3], [4]. These reserves are typically controllable loads like EV charging, space heating, and hot water boilers,

which can be used as energy storages. The benefit for distribution system operator (DSO) and electricity market is indisputable and thus it can be expected that a customer would get investment support and supporting tariffs [5].

This cooperation between service provider and end customer is called demand response (DR). It is defined as change in electricity usage by the end user from their general consumption pattern in response to changes in the price of electricity [6]. The most basic type of price-based demand response is based on time of use (TOU) rates, which have on-peak, off-peak and shoulder rate periods [3]. A day-ahead-pricing (DAP) provides better flexibility and thus is often used when implementing DR for house energy management systems (HEMS) [3], [4].

A case-study of this work also combines use of DAP tariff with use of energy storage system (ESS). The household of this case-study is however equipped with photovoltaic (PV) installation and therefore a production following has become priority. The key incentive behind designed system was low-cost and accessibility of required components.

For these reasons a hot water boiler is chosen as energy storage. It is more common to analyze systems using battery storage systems assuming significant decrease in its prices in the near future [3], [4]. Hot water boiler can, on the other hand, provide sufficient capacity for reasonable price. The downside of heat storages over battery storages is its limited versatility. It can be, however, justified in cases when it is used for space heating. Utility use of hot water is not considered as the analysis would become, due to non-deterministic behavior of residents, too complex.

A thermal model of the household is implemented in chapter 3, in order to carry out simulations of this case-study. The algorithm, that implements production following and DAP based DR, is presented in chapter 4. Analysis of case-study that focuses on factors like size of energy storage and size of photovoltaic installation is presented in chapter 5.

This work also comprises of laboratory development presented in chapter 6. The idea is to have laboratory capable of household emulation using real appliances in real-time. A Matlab / Simulink is used for implementation of all algorithms and thus a corresponding interface to laboratory equipment needs to be programmed. The main part of this work is implementation of load control using wireless Z-wave interface and implementation of measurement system using dSpace. This hardware in the loop (HIL) system can easily execute algorithms developed with household models as it uses same Simulink environment.

Chapter 7 discusses simulation results proposing possible improvements together with future development of the laboratory. The final chapter concludes the thesis and the most important results are recapped.

2. AC MICROGRID INTRODUCTION

This thesis participates to bigger Smart Grid Energy Market (SGEM) project, specifically Work Package 4 (WP4). The initial idea of AC Microgrid laboratory was specified by schematics depicted in Figure 2.1 below. The core of the laboratory, called AC Microgrid panel, comprises of electrical installation of typical household with Smart Meter, circuit breakers, current sensors and contactors to allow control and automation by house energy management system (HEMS). Moreover, a second contactor is installed in series with the first one to allow emulation of household's environment. These contactors will be controlled by dSpace emulating e.g. behavior of residents. The AC Microgrid panel interfaces with loads, energy storages, DG, dSpace and HEM PC.

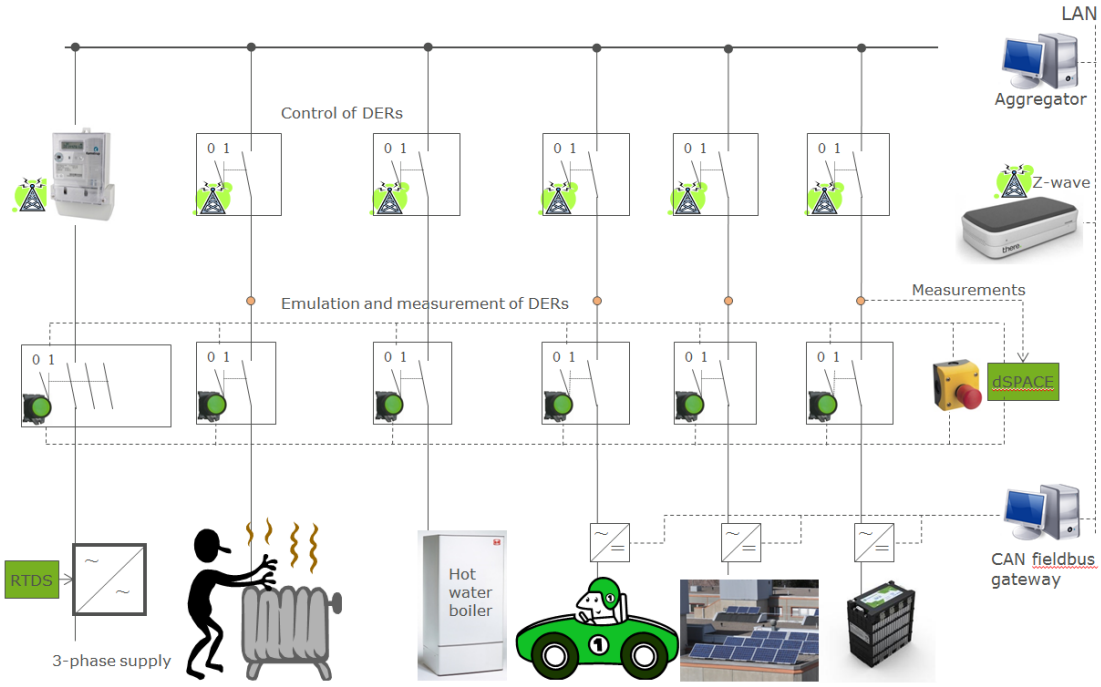


Figure 2.1. *Initial schematics of AC Microgrid laboratory*

This AC Microgrid laboratory is planned to be connected to other laboratory in order to emulate its cooperation with distribution network. Schematic of this setup is captured in Figure 2.2 below. An aggregator provides e.g. hourly prices that are processed by Tertiary controller that implements scheduler. Generated schedules are then interpreted by Secondary controller that controls appliances in real-time. The last, primary controller is integrated within appliances e.g. as a part of power electronics and therefore doesn't concern this work.

Operation of house energy management system can be affected by Distributed Intelligent Electronic Device (DIED) e.g. in emergency situations to mitigate network's disturbances. DIEDs represent the interface between sensors/actuators and monitoring and control application in the distribution grid and in the DG units of the prosumers. In particular, they may represent the interface to recloser in distribution feeders, distributed generators and distributed power quality meters. DIEDs are controlled by Substation Automation Units (SAUs) that directly interface with Supervisory Control And Data Acquisition (SCADA) Centre. These units are in charge of processing data at primary or secondary substation level. [7]

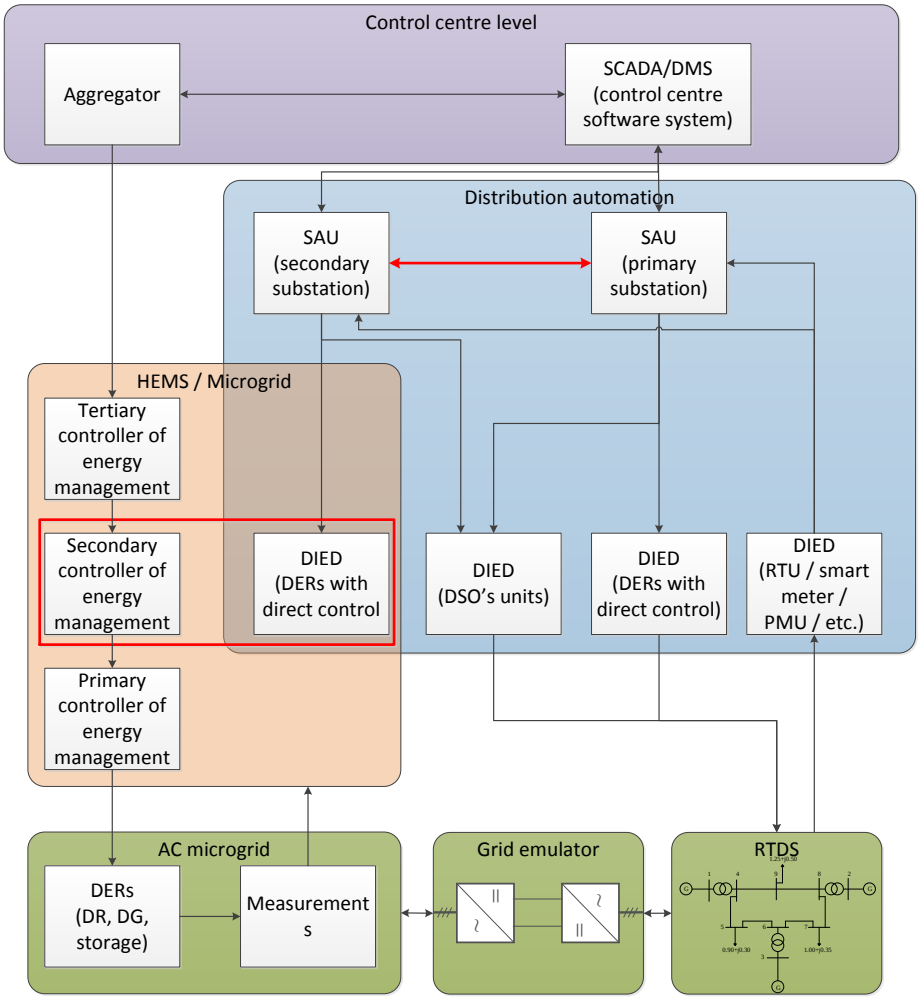


Figure 2.2. *Integration of AC Microgrid to bigger project*

AC Microgrid depicted in green color represents physical laboratory that comprises of distributed energy resources (DERs) like distributed generation (DG), energy storage system (ESS) or another device capable of participation in demand response (DR). Measurements of power-flows and temperatures are provided to HEM PC required by Secondary controller. This laboratory will be powered by Grid emulator to provide voltage and frequency disturbance in order to emulate any scenario of distribution network. This Grid emulator will be controlled by Real Time Digital Simulator (RTDS) executing model of distribution network.

3. MODEL OF THE HOUSEHOLD

In this chapter, a model of household is described that is needed for case-study of home energy management system's (HEMS) algorithm. The design of this model has been done in correspondence with equipment in laboratory in the Department of Electrical Engineering at Tampere University of Technology that will be gradually replacing parts of this model.

All appliances that can be found in a typical household can be classified into three classes regarding load's deferrable potential. A first one comprises of loads whose load profile is predictable when turned on. This class includes, for example dishwasher, electric vehicle (EV) etc. Appliances of second class have predictable load profile except for an unknown duration. Appliance of this class can be for example lamp, television set, computer, etc. These are extremely time sensitive loads, and so is their electricity demand. Therefore, this class doesn't provide much controllable capacity for demand response (DR). Last, but not least, the third class of loads includes, for example, thermostatically controlled appliances. These loads can be both interrupted and turned on earlier.

Most load control implementations are controlling appliances of this class to get reasonable controllable capacity without affecting comfort of residents [8]. First class has good DR potential too, but utilizing these appliances is more challenging for at least two reasons. First, the appliance needs to be equipped with a remote control capability that is needed to relay control signal from HEMS to e.g. dishwasher to start its program or to EV to adjust its charging power. Secondly a load profile needs to be known in order to schedule the start of devices operation at optimal time. Therefore, for easier demonstration of load control implementation, only third class of appliances has been used.

A hot-water tank, that can store heat energy equivalent to more than 13kWh, connected in closed-loop system with pump and radiator has been modelled as this system is planned to be built later in the laboratory and belongs into load category that is most suitable for load deferring. It is important to note, that designed use of heat stored in this system is only for space heating. A model of hot-water tank, often referred to as water boiler is described in chapter 3.2 and the radiator with pump controller is described in chapter 3.3. Parallel to a radiator a direct electric space heater has been implemented to assist closed loop system in situations when 4kW rated radiator wouldn't be enough to satisfy required temperature. This heater will be represented by 6kW sauna heater in laboratory setup. The model of whole household that has been developed for case study-purposes is depicted in Figure 3.1 below.

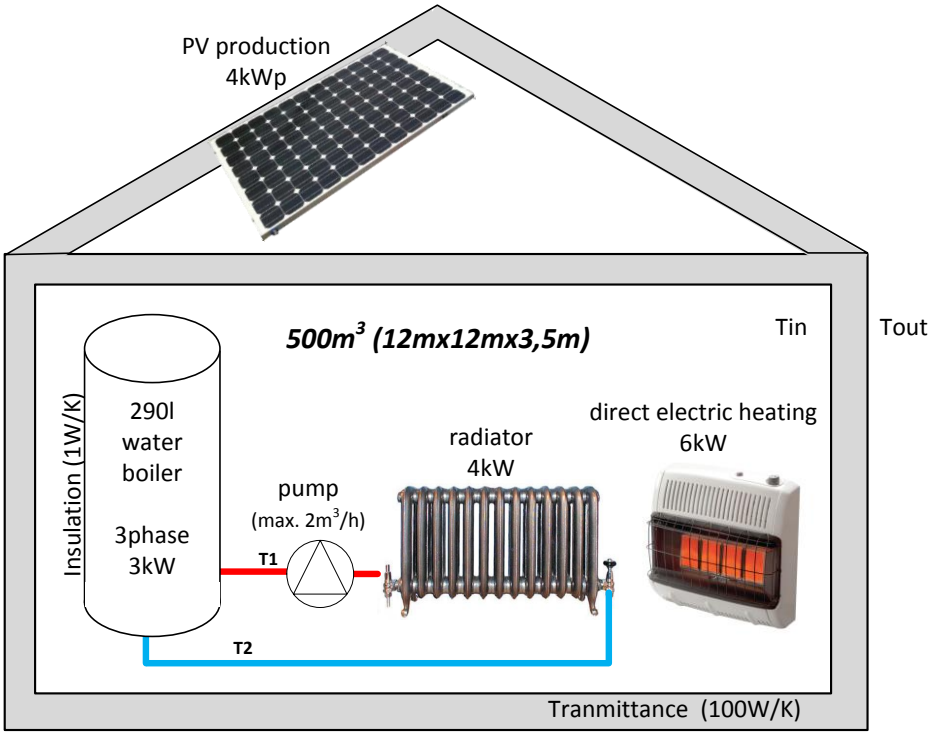


Figure 3.1. *Topology of household used for case-study*

The selection of household's parameters captured in Figure above has been inspired by typical detached family house with local photovoltaic (PV) production [9], [10] and [11]. This model has been encapsulated into subsystem that is shown in Figure 3.2 below. It has four inputs of which three are control inputs. Two are binary that switch relay for water boiler heating and electric space heater and one is continuous that controls heat dissipated by radiator. The last, fourth, input provides model with information about outdoor temperature.

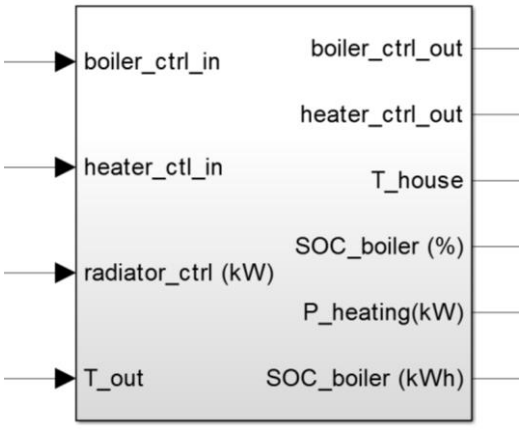


Figure 3.2. *Household model*

The model outputs boiler's and heater's control signals that can differ from input signals by action of e.g. over-heat protection. Third output provides indoor temperature of the household that is needed by real-time controller, which implementation is explained in chapter 4.2. Last three outputs give state of charge (SOC) of water boiler and power

used for heating. Photovoltaic (PV) production didn't need to be included as it is represented by data stream that does not require any processing.

Figure 3.3 below shows *Simulink* model of *Household model* subsystem. The most complex subsystems of this model are *House*, *WaterBoiler* and *pump-controller-NR* that are described in chapters 3.1, 3.2 and 3.3, respectively.

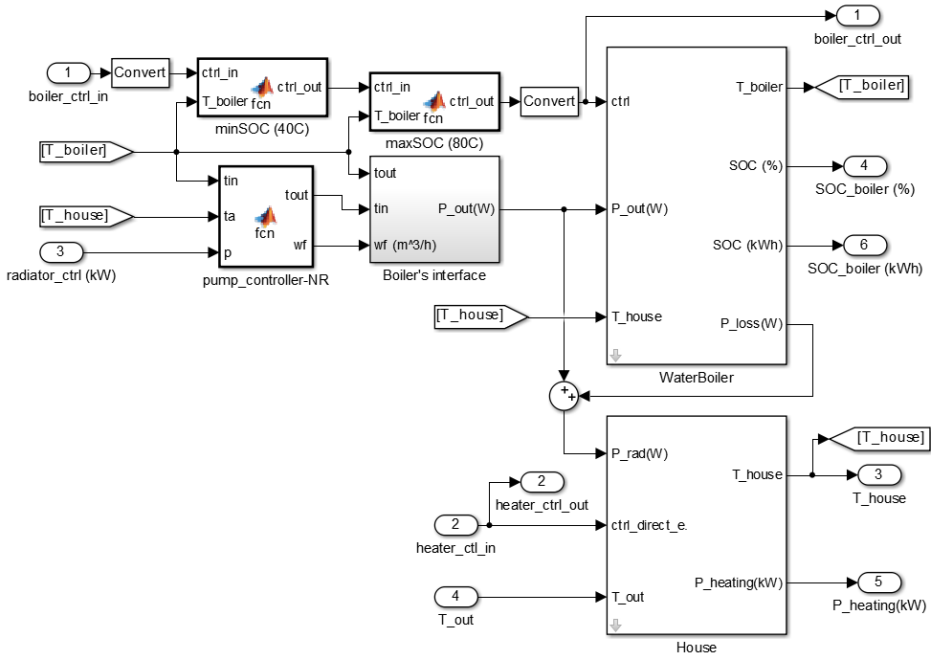


Figure 3.3. *Simulink model of household*

Couple of systems connected in series called *minSOC(40C)* and *over maxSOC(80C)* ensures that the temperature of water boiler stays within 40°C and 80°C temperature range that translates into 0-100% SOC range.

```
function ctrl_out  = minSOC(ctrl_in,T_boiler)

persistent forced;
if(isempty(forced))
   forced = 0;
end

% temp. below which heating will be forced
hyst_low = 40;
% temp. above which heating signal wont be
% affected
hyst_hi = 41;

if(T_boiler < hyst_low)
     forced = 1;
end

if(T_boiler > hyst_hi)
     forced = 0;
end

ctrl_out = (ctrl_in || forced);

end
```

```
function ctrl_out  = maxSOC(ctrl_in,T_boiler)

persistent stopped;
if(isempty(stopped))
   stopped = 0;
end

% temp. below which heating wont be affected
hyst_low = 79;
% temp. above wehich heating will be stopped
hyst_hi = 80;

if(T_boiler < hyst_low)
     stopped = 0;
end

if(T_boiler > hyst_hi)
     stopped = 1;
end

ctrl_out = (ctrl_in && ~stopped);

end
```

Figure 3.4. *Implementation of minimum (left) and maximum (right) SOC controller*

A hysteresis had to be implemented in order to prevent oscillations as can be seen from implementation shown in Figure 3.4.

The last system called *Boiler's interface* calculates dissipated heat in Watts based on water flow *wf* in closed loop system, temperature at the input t_{in} and output t_{out} terminals of boiler, by implementing

$$P_{out} = wf \cdot c_w \cdot \frac{10^6}{3600} \cdot (t_{in} - t_{out}) , \tag{3.1}$$

where c_w is specific heat capacity of water, that is 4.184 $J \cdot kg^{-1} \cdot K^{-1}$. Calculated power P_{out} that we achieved should be same as *radiator_ctrl* input of *pump-controller-NR* as long as the power rating of radiator and pump is sufficient.

3.1 Model of the house

Model of the house implements two heat sources: electric heater and radiator of closed loop system. This model does not implement all factors that have an influence on ambient temperature. Instead a simplified one room model has been used that accounts only for passive losses caused by non-ideal insulation. Therefore losses caused by greenhouse effect, opened windows or heating by residents and appliances losses are not considered. Therefore the simulation results should be taken only as an indication of real-life performance. This model is based on equation (3.2) where T_{init} is initial temperature that is set to desired indoor temperature.

$$T_h = T_{init} + \frac{Q_{tot}}{C} = T_{init} + \int \frac{P_{tot}(t)}{C} dt \tag{3.2}$$

Total power P_{tot}, as can be seen in (3.3), comprises of two heating sources, electric heater $P_{el.}$ and a radiator $P_{rad.}$ and losses represented by P_{loss}. C, the heat capacity is composed of two parts, first represents heat capacity of air C_{air}, and the other heat capacity of the house itself C_h that encompass walls, furniture, etc.

$$T_h = T_{init} + \int \frac{P_{el.}(t) + P_{rad.}(t) - P_{loss}(t)}{C_{air} + C_h} dt \tag{3.3}$$

Heat capacity of air C_{air} is calculated using (3.4) where specific heat capacity of air c_{air} is 1,007 $kJ \cdot g^{-1} \cdot K$ and mass of air m_{air} is calculated using known density of air - $1204 g \cdot m^{-3}$ and volume of the house V_h [12].

$$C = c \cdot m \tag{3.4}$$

The power used by electric heater $P_{el.}(t)$ can be expressed as multiplication of its rated power $P_{el.}$ and its driving binary signal $ctrl_{el.}(t)$. Insulation losses are calculated using thermal transmittance of insulation U, its area A and temperature difference between house's temperature T_h and outdoor temperature T_o as can be seen in (3.5) below.

$$T_h = T_{init} + \int \frac{P_{el.} \cdot ctrl_{el.}(t) + P_{rad}(t) - U \cdot A\big(T_h(t) - T_o(t)\big)}{(V_h \cdot 1204 \cdot 1{,}007) + C_h}\, dt \quad (3.5)$$

Figure 3.5 below captures Simulink implementation of above described equation (3.5). A discrete integration has been used as the code will be executed in real-time with constant sample time when compiled to dSpace.

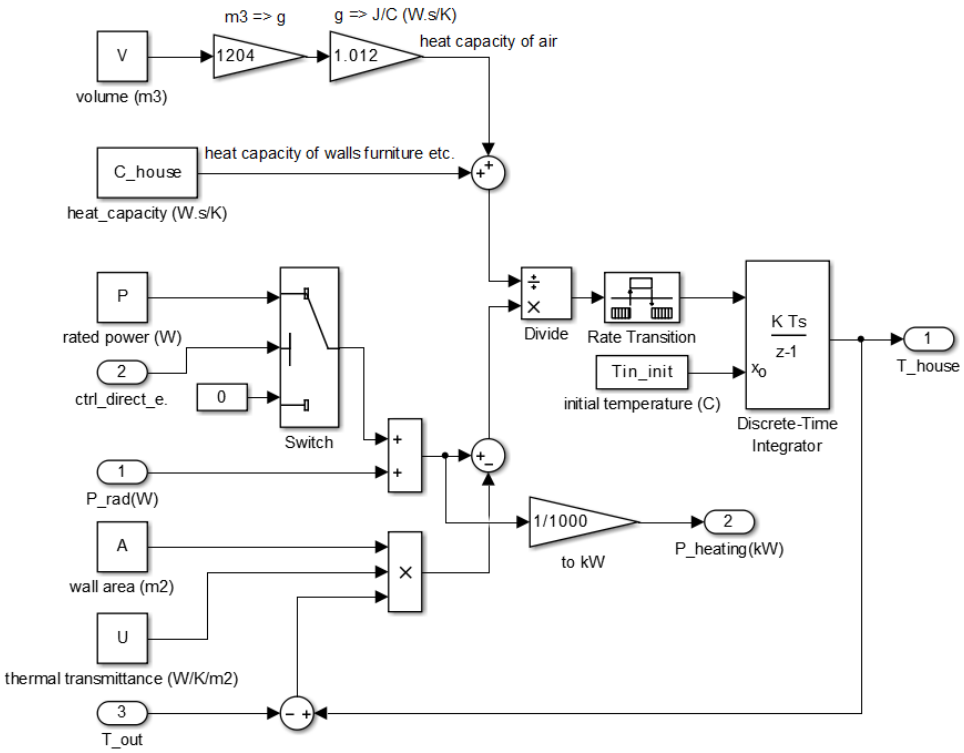


Figure 3.5. *Model of the house*

3.2 Model of the hot water boiler

Model of water boiler is similar to the model of house described in previous chapter. In this case the system has only one heat source – the resistor of water boiler with rated power P_{in} and two outputs. First represents heat going to radiator P_{out} and the other represents insulation losses P_{loss} as can be seen in equation (3.6). Both outputs are later added together and used as heat source for model of the house as can be seen in Figure 3.3.

$$T_b = T_{init} + \int \frac{P_{in} - P_{out} - P_{loss}}{C_w} dt \qquad (3.6)$$

Heat capacity of water C_w is calculated using (3.4) where specific heat capacity of water c_w is 4,187 $kJ \cdot g^{-1} \cdot K$ and mass of water m_w is calculated using known density of water $1000g \cdot l$ and volume of the boiler V_b in liters. [12].

$$T_b = T_{init} + \int \frac{P \cdot ctrl(t) - P_{out}(t) - U \cdot A(T_b(t) - T_h(t))}{(V_b \cdot 1000 \cdot 4{,}187)} dt \qquad (3.7)$$

The loss caused by non-ideal insulation P_{loss} is calculated using difference between temperature of boiler T_b and house T_h, thermal transmittance of insulation U and surface area of boiler A. Calculation of SOC has been added as it is needed for real-time control and evaluation of algorithm's performance. The SOC is defined as ratio of stored Q_s and rated Q_r capacity and can be simplified into ratio of temperatures as shown in (3.8) and (3.9). [13]

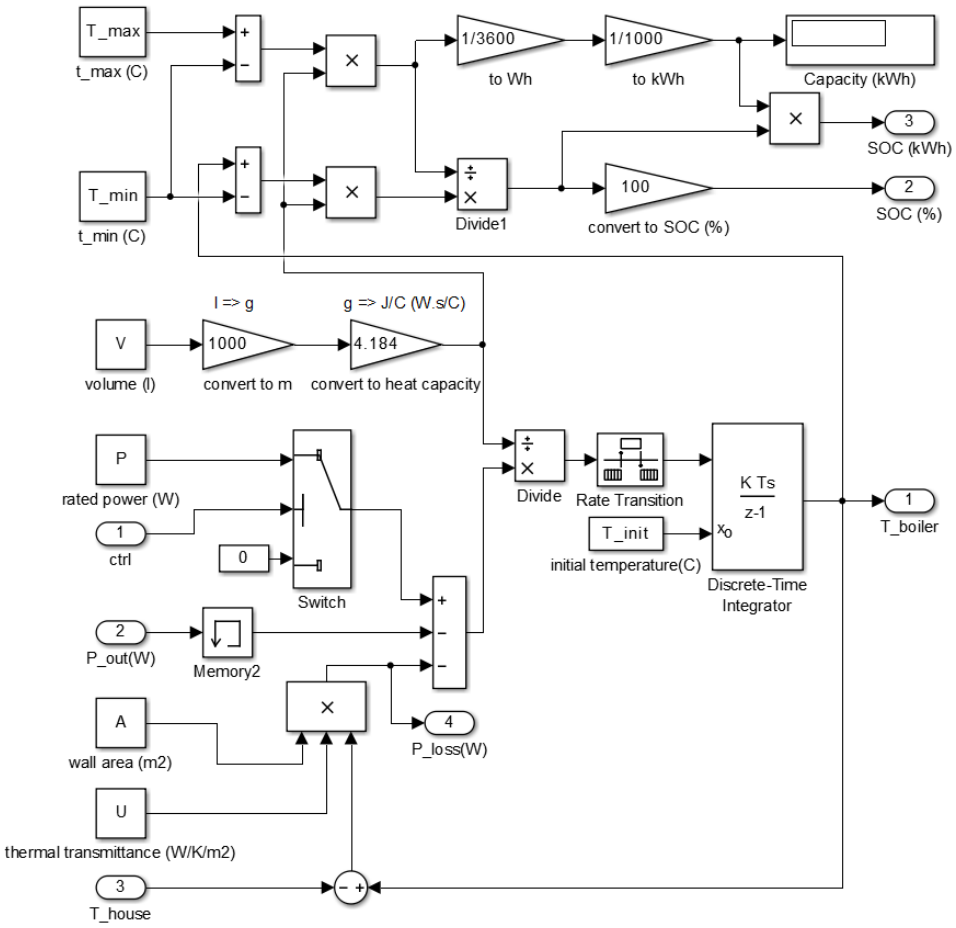


***Fig 3.6**. Model of water boiler*

$$Q = \Delta T \cdot C_w \cdot m_w \tag{3.8}$$

$$SOC = \frac{Q_s}{Q_r} \cdot 100 = \frac{\Delta T_s \cdot C_w \cdot m_w}{\Delta T_r \cdot C_w \cdot m_w} \cdot 100 = \frac{T_b - T_{min}}{T_{max} - T_{min}} \cdot 100 \tag{3.9}$$

Where T_{max} and T_{min} specify boiler's operational temperature that has been set to 80°C and 40°C, respectively. The unit purchased for laboratory of this project has 290 litters volume that gives storage capacity equivalent to 13.48kWh. Implementation of this model can be seen in Figure 3.6 above.

3.3 Model of the radiator

A mathematical model of radiator was created in order to calculate water-flow that will result in radiator dissipating required amount of heat. In presented case-study scenario, a calculated power flow is used to generate driving signal for variable-speed pump. This approach has been used as the modelled house comprises of only one room. However, if house with multiple rooms would have been modelled with requirement to set individual heat dissipation, then each radiator would have to have its own pump and pipeline. This wouldn't be much sensible solution and therefore installing controllable valves on each radiator would be preferred. However, even in this scenario a derived water-flow could be used with small modification. The calculated flow would be used to determine valve's position as it would change the ratio of water flows between radiators. Controller of this pump containing radiator's model is described in this chapter.

Inputs of controller are ambient temperature at the input of the radiator T_{r_in} , temperature of house T_h, and the heat to be dissipated by the radiator *radiator_ctrl*. The controller has two outputs, water flow *wf* in cubic meters per hour m^3/h and temperature of water returning from radiator T_{r_out}. A Matlab function has been used to implement controller into Simulink model as can be seen in Figure 3.7 below.

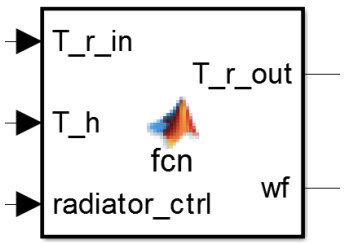


Figure 3.7. *Radiator controller*

Parameters of the radiator chosen for the case study are listed in Table 3.1 below.

Table 3.1 *Parameters of radiator*

symbol	***name***	***value***	***unit***
P_N	radiator's rated power	4000	W
$t_{r\_out,N}$	radiator's exhaust temperature	60	$°C$
$t_{r\_in,N}$	radiator's intake temperature	80	$°C$
$t_{h,N}$	house temperature	20	$°C$
$c_{w,N}$	specific heat capacity - water	4,184	$J \cdot kg^{-1} \cdot K^{-1}$
n	radiator's heat emission factor	1,33	$-$

The model of radiator is based on the first thermodynamic law, which says that the change of energy ΔE of the system is equal to heat transferred to the system Q minus work W done by it.

$$\Delta E = Q - W \tag{3.10}$$

In case of radiator the system operates in steady-state and therefore ΔE is equal to zero. Moreover no work is done, so only heat transfer is left that comprises of heat transferred to and from the system, represented by Q_{in} and Q_{out}, respectively. Therefore an expression (3.10) can be written as (3.11).

$$Q_{out} - Q_{in} = 0 \tag{3.11}$$

When heat of Joule's first law (3.12) is substituted into (3.11) then a balance equation (3.13) can be derived that will be used as a balance equation in Newton-Raphson iterative algorithm.

$$Q = P \cdot t \tag{3.12}$$

$$P_{out} - P_{in} = 0 \tag{3.13}$$

As the heat transferred from the system Q_{out} is input variable of the controller, only input heat Q_{in} has to be expressed. The performance of a radiator can be described using radiator's performance under nominal conditions P_N and heat emission factor n. This n factor represents characteristic emission curve of radiator and is typically equal to 1.33 [14]. Input heat of radiator can be expressed as (3.14) and when is first Joule's law applied an input power can be expressed - (3.15).

$$Q_{in} = Q_N \left(\frac{LMTD}{LMTD_N}\right)^n \tag{3.14}$$

$$P_{in} = P_N \left(\frac{LMTD}{LMTD_N}\right)^n \tag{3.15}$$

Abbreviation LMTD in equations (3.14) and (3.15) stands for *logarithmic mean temperature difference* and notation N for operation at nominal conditions that are listed in Table 3.1.

$$LMTD = \frac{\Delta t_A - \Delta t_B}{ln\frac{\Delta t_A}{\Delta t_B}} \tag{3.16}$$

LMTD can be calculated according (3.16) where Δt_A stands for difference between input temperature of water t_{r_in} and output temperature of air t_{h_out}. Similarly Δt_B is difference between output temperature of water t_{r_out} and input temperature of air t_{h_in} as can be seen in Figure 3.8. It is common practice to simplify the solution by assuming that the ambient temperature is same on the output as on the input. [15] Therefore a simplification and change in notation ca be done as shown in (3.17)

$$\begin{aligned} t_{h\_in} = t_{h\_out} = t_h \\ t_{r\_in} = t_{in} \\ t_{r\_out} = t_{out} \end{aligned} \tag{3.17}$$

Applying notation (3.17) results in simpler expression of Δt_A and Δt_B as shown in (3.18).

$$\begin{aligned} \Delta t_A = t_{r\_in} - t_{h\_out} = t_{in} - t_h \\ \Delta t_B = t_{r\_out} - t_{h\_in} = t_{out} - t_h \end{aligned} \tag{3.18}$$

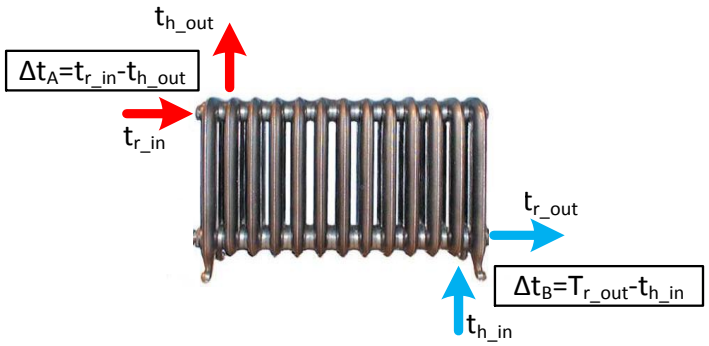


Figure 3.8. *Input and output temperatures of radiator's model*

By using expression in (3.18) an equation (3.16) can expressed as shown in (3.19).

$$LMTD = \frac{t_{in} - t_{out}}{ln\frac{t_{in} - t_h}{t_{out} - t_h}} \tag{3.19}$$

Logarithmic mean temperature difference at nominal conditions of radiator can be expressed while initializing controller and can be therefore handled as a constant for all simulations.

$$LMTD_N = \frac{t_{in,N} - t_{out,N}}{ln\frac{t_{in,N} - t_{h,N}}{t_{out,N} - t_{h,N}}} = \frac{80 - 60}{ln\frac{80 - 20}{60 - 20}} = 49{,}32 \tag{3.20}$$

By substituting (3.19) and (3.20) into (3.15) and (3.13) power balance equation (3.21) can be expressed as a function of output temperature of water returning from tor t_{out}.

$$F(t_{out}) = P_{r\_out} - P_N \left(\frac{t_{in} - t_{out}}{ln\left(\frac{t_{in} - t_h}{t_{out} - t_h}\right) \cdot LMTD_N} \right)^n = 0 \tag{3.21}$$

In order to calculate t_{out} a Newton-Raphson iterative method has to be used. This method requires partial derivative of balance equation (3.21) as can be seen from equation (3.22).

$$t_{out,new} = t_{out,old} - \frac{F(t_{out,old})}{\frac{\partial F(t_{out,old})}{\partial t_{out}}} \tag{3.22}$$

$$\frac{\partial F(t_{out})}{\partial t_{out}} = \\ = \frac{P_N \cdot n \cdot LMTD_N \left((t_{out} - t_h) \cdot ln\left(\frac{t_h - t_{in}}{t_h - t_{out}}\right) + t_{out} - t_{in} \right) \left(\frac{t_{in} - t_{out}}{LMTD_N \cdot ln\left(\frac{t_h - t_{in}}{t_h - t_{out}}\right)} \right)^{n+1}}{(t_{out} - t_{in})^2 (t_{out} - t_h)} \tag{3.23}$$

When an iterative algorithm provides output temperature with desired precision a water flow, a driving signal for water pump, can be expressed by using (3.24).

$$P_{out} = P_{in} = \dot{m}_w \cdot c_w (t_{in} - t_{out}) \tag{3.24}$$

Where $\dot{m}_w$ is mass flow in kilograms per second and c_w is specific heat capacity of water [13], [14]. By modifying (3.24) a water flow wf in cubic meters per hour can be expressed.

$$\dot{m}_w = \frac{P_{out}}{c_w \cdot (t_{in} - t_{out})} \ (kg \cdot s^{-1}) \tag{3.24}$$

$$wf = \dot{m}_w \cdot \frac{3600}{10^6} = \frac{P_{out}}{c_w \cdot (t_{in} - t_{out})} \cdot \frac{3600}{10^6} (m^3 \cdot h^{-1}) \tag{3.25}$$

The controller is curtailing maximum pump speed to represent its physical limitation. A maximum speed of $2m^3 \cdot h^{-1}$ has been set as it is maximum speed of pump used in laboratory setup. The iterative Newton-Raphson calculation is limited to 10 iterations with precision set to 0.1°C. To get this precision 3 to 7 iterations is required.

4. PRODUCTION FOLLOWING

In this chapter, a production following algorithm is described. The potential of this algorithm in not only beneficial to residents by savings on electricity bills, but it is also beneficial for the electricity network. The biggest benefit of presented implementation is load deferring from peak hours to off-peak hours. Only this has positive impact on quality and reliability of electricity distribution [16], [17]. Moreover a modification of *real-time controller* could be done to participate in real time pricing (RTP), emergency demand response (EDR) in reactive power control. It would however require real-time communication with e.g. aggregator – a server located between Distribution System Operator (DSO) and Home Energy Management System (HEMS) computer. For this reason an implementation of this control hasn't been included in this work.

An implementation described in this chapter is controlling heating loads in order to reduce electricity bills without effect on house's temperature. As can be seen in Figure 4.1 below, the algorithm is divided into two parts – *Scheduler* and *Real-Time control* that are described in chapters 4.1 and 4.2, respectively.

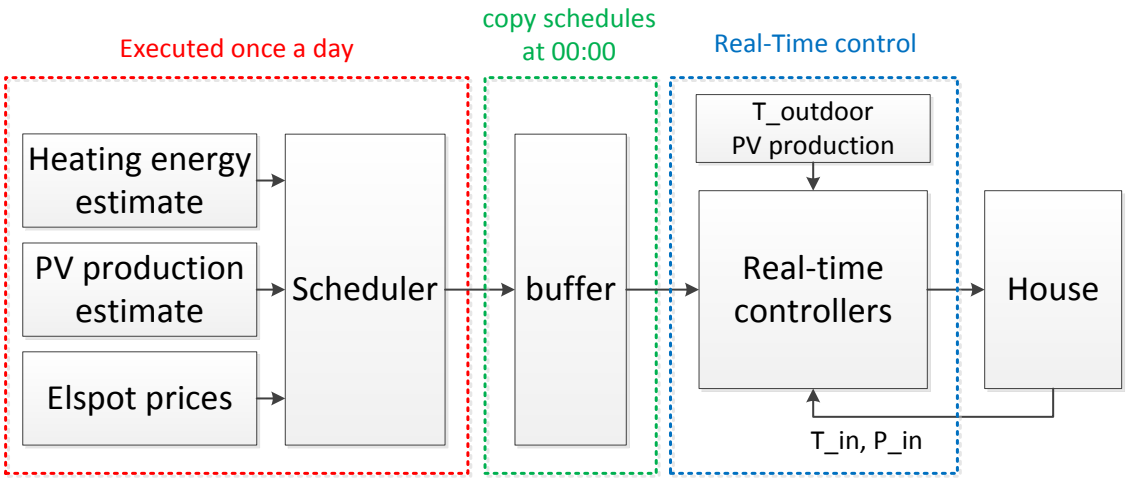


***Figure 4.1**. Algorithm's flowchart*

4.1 Scheduler

The scheduler has been implemented in order to carry out schedules for charging and discharging energy storage system (ESS), in this case water boiler. These schedules are prepared once a day for upcoming day starting at midnight. It is important to realize that generated schedules are representing volumes of energy to be applied in each hour. Therefore the unit of measure the scheduler is operating with is energy in kilowatt-hours. The algorithm's objective is to ensure enough free capacity in ESS for local PV

production and to charge it from network when the price of electricity is lowest. Therefore a load shifting is achieved while keeping reserves to store energy of local production.

As described in chapter 3 a model of household includes two heating units, electric space heater and radiator connected with water boiler in closed-loop system. Therefore three schedules needs to be prepared one for each unit: water boiler, radiator and electric space heater. It should be noted, that radiator is continuously controlled load as the speed of pump can be controlled.

4.1.1 Input data

Before a scheduler can be executed, input data has to be prepared. Prices of day-ahead market provided by Nord Pool are of one hour resolution. This is the reason why the scheduler operates with row vectors of length 24. Therefore the other inputs have to be resampled to this resolution. Forecast of PV production and outdoor temperature is obtained from database of measurements made by weather station at Tampere University of Technology [18]. Therefore an ideal forecast can be achieved that is very convenient as a proper behavior of algorithm can be verified. However, to produce realistic simulation results an error can be added to both PV production estimate and outdoor temperature according (4.1) and (4.2), respectively.

$$P_{PV\&err} = \left(\frac{err_p}{50} \cdot rand - \frac{err_p}{100} + 1\right) \cdot P_{PV} \qquad P_{PV\&err} \leq P_{PV\_rated} \tag{4.1}$$

Where err_p is introduced error in percentage, $rand$ is random number on 0-1 interval with uniform distribution, P_{PV} is PV production without added error and P_{PV_rated} is rated power of PV installation. This implementation adds error with maximum power limit. However, better implementation would limit PV production to maximum theoretical production that would require model comprising of tilt and orientation of PV modules and information about day of a year. Hence, for its complexity, presented implementation is used.

$$T_{out\&err} = T_{out} - \frac{err_c}{2} + err_c \cdot rand \qquad \frac{\Delta T_{out\&err}}{\Delta t} < 2 \tag{4.2}$$

The error added to temperature is done according (4.2), where err_c stands for error in degrees Celsius, $rand$ for random number uniformly distributed over 0-1 range and T_{out} and $T_{out\&err}$ for outdoor temperature without and with introduced error, respectively. In order to represent more realistic weather forecast a gradient limit was implemented to two degrees Celsius per hour, that according [19] results in estimate closer to real weather forecast.

The forecast of outdoor temperature is needed to make an estimate of heating requirements as can be seen in Figure 4.2. A power loss is calculated using (4.3)

$$P_{he} = \frac{(T_{in} - T_{out\&err}) \cdot U \cdot A}{1000} \ (kW), \tag{4.3}$$

where U is thermal transmittance of wall, A is its area, T_{in} is indoor temperature and $T_{out\&err}$ is outdoor temperature. This P_{he} loss can be considered constant over one hour as the input data is of one hour resolution. Therefore, expressing heat that has to be generated, in order to keep indoor temperature constant, can be done as follows.

$$E_{he} = P_{he} \cdot 1 = \frac{(T_{in} - T_{out\&err}) \cdot U \cdot A}{1000} \ (kWh) \tag{4.4}$$

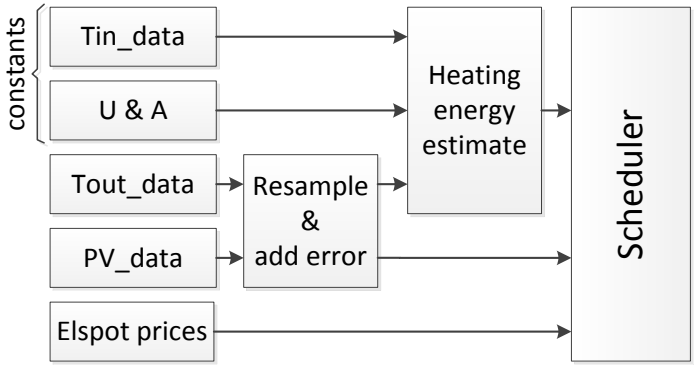


Figure 4.2. *Input data processing*

4.1.2 Scheduler

Two scheduling strategies were implemented where the second one is an extension of the first one. Therefore the first one is explained in this chapter and the difference between them is covered in chapter 4.1.3. An example of schedule build-up is shown parallel to its description using data from May 5th 2012.

First a heating demand is calculated as difference between heating energy estimate and PV production estimate provided by (4.4) and (4.1), respectively. As can be seen on example shown in Figure 4.3 a *heating demand* can be negative. It can happen in mid-day when a PV production is higher than heating requirement. At this time an excessive production could be either stored or sold to network. It is desired to store all production as long as there will be use for it. However if there is no estimated heating requirement, especially in the summer, then it is better strategy to sell this energy. Moreover the water boiler insulation is not ideal, as described in chapter 3.2, and as it is placed within household, it would heat the house even-though the ambient temperature might already be above desired temperature.

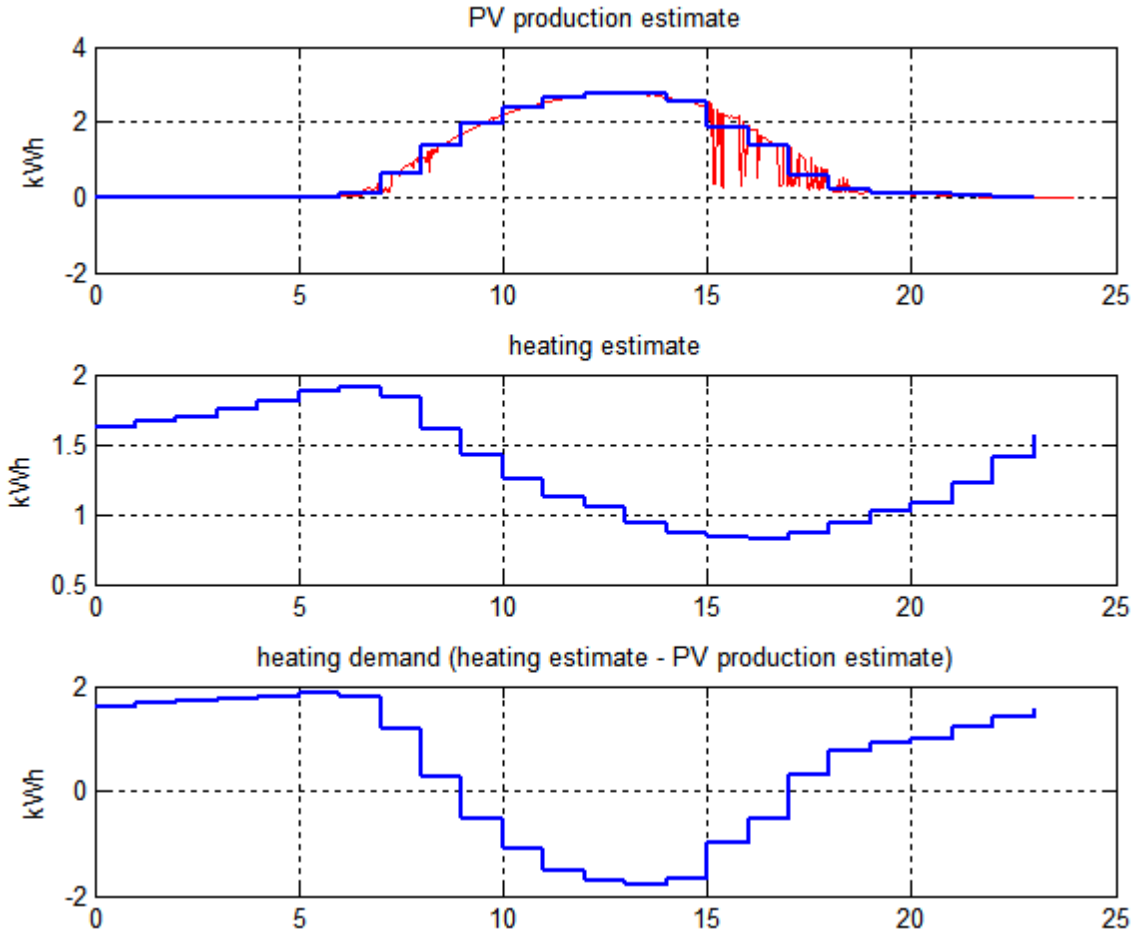


Figure 4.3. *Heating demand*

This is the reason why the state of charge (SOC) at the end of day should stay below specified value called *SOCEndMax*. This constraint is defined as a fraction of heating estimate - (4.5).

$$SOCEndMax = 0{,}55 \cdot E_{he} \tag{4.5}$$

This expression is assuming that the heating estimate does not differ much between two consecutive days. Ideally, this constraint would be expressed using weather forecast of following day. This approach would result in better *SOCEndMax* constraint but for increased complexity a simpler approach was used.

This is, however, not the only constraint. The SOC is not allowed to go out of 0-100% range. This range can be curtailed in order to provide safety margins that would compensate for error in input data. These marginal parameters are specified separately for lower and upper bounds of the range as shown in (4.6). Parameters in these equations were chosen based on simulations presented in chapter 5.

$$\begin{gathered} SOCminMargin < SOC < (BoilerCapacity - SOCmaxMargin) \\ SOCminMargin = 0{,}01 \cdot BoilerCapacity \\ SOCmaxMargin = 0{,}02 \cdot BoilerCapacity \end{gathered} \tag{4.6}$$

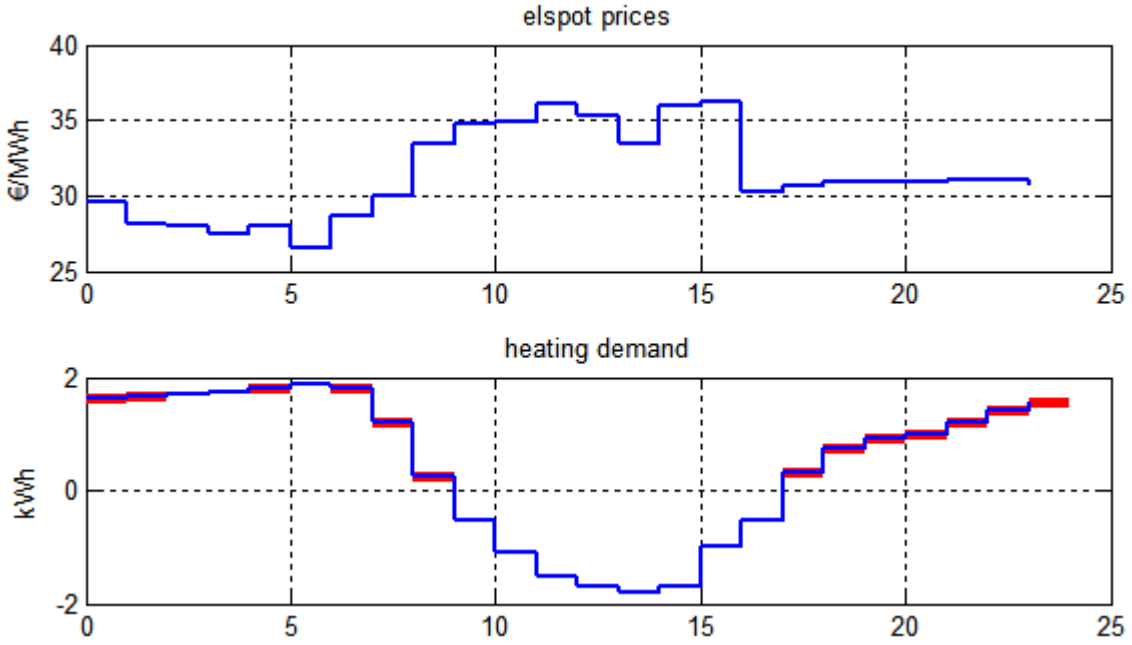


Figure 4.4. *Selection of deferred hours*

The first step in schedule generation is selection of hours at which a heating demand will be covered by energy stored in ESS. In order to do that, the prices of day-ahead market are used [20]. As can be seen in Figure 4.4, hours marked by red line were chosen for discharging ESS that essentially define schedule for radiator. They were chosen for the highest prices, while heating demand was positive. The number of hours to be scheduled is decided by energy to be discharged (*DischargeE*) – one of the state variables of algorithm. Similarly a schedule for charging ESS is based on total energy to be stored (*ChargeE*) – second state variable.

In order to get values of state variables that result in best schedule without constraint violation a search algorithm is implemented. The precision of generated schedules does not have to be perfect as the precision of input data is already affected by forecast error. Instead, a light-weight implementation is preferred as it is intended to be executed on low-cost HEM computer. Therefore the search algorithm can e.g. assign ten possible values to *ChargeE* and ten to *DischargeE* that would result in hundred possible combinations. Simulation results, presented in chapter 5.3, indicate that discretization of state-variables to 36 states gives best ratio of monetary savings and memory footprint. In order to assign values to state variables a minimum and maximum value has to be specified. Therefore each state variable has three parameters that specify state-space to be searched as can be seen in Table 4.1 below. The minimum energy to be charged and discharged from ESS is set to zero to ensure that even when there is very little use for ESS the best solution wouldn't be missed. The maximum energy to be charged has been set to double of the energy that can EES store. Therefore two full cycles of ESS are possible within one day.

As the search algorithm goes through all combinations of state variables a non-feasible schedule can emerge. It can happen e.g. when *DischargeE = 0 and ChargeE = 2·boilerCapacity* that would result in SOC going over 100%. Therefore a constraint (4.7) would be violated and this state of state-space would be ignored.

Table 4.1 *State variable search parameters*

name	***value***	***unit***
$DischargeEmin$	0	kWh
$DischargeEmax$	$2 \cdot boilerCapacity$	kWh
$DischargeEstep$	$\frac{DischargeEmax - DischargeEmin}{28}$	kWh
$ChargeEmin$	0	kWh
$ChargeEmax$	$2 \cdot boilerCapacity$	kWh
$ChargeEstep$	$\frac{ChargeEmax - ChargeEmin}{28}$	kWh

Now, when a state-space at which is the search done is defined, the process that happens for each combination of state variables will be explained. This process comprises of three steps as can be seen from flowchart in Figure 4.6. First, constraints are checked. SOC schedule has to be generated as all constraints have been defined relative to SOC as seen in (4.5) – (4.7). The Figure 4.5 below captures one of such schedules in kWh where the maximum SOC is limited to 100% that translates to 13.48kWh.

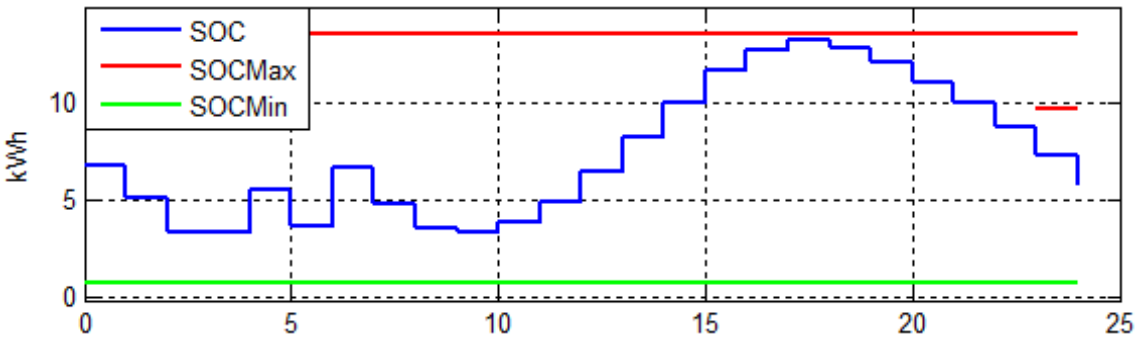


Figure 4.5. *SOC in kWh with SOCmax / SOCmin constraints*

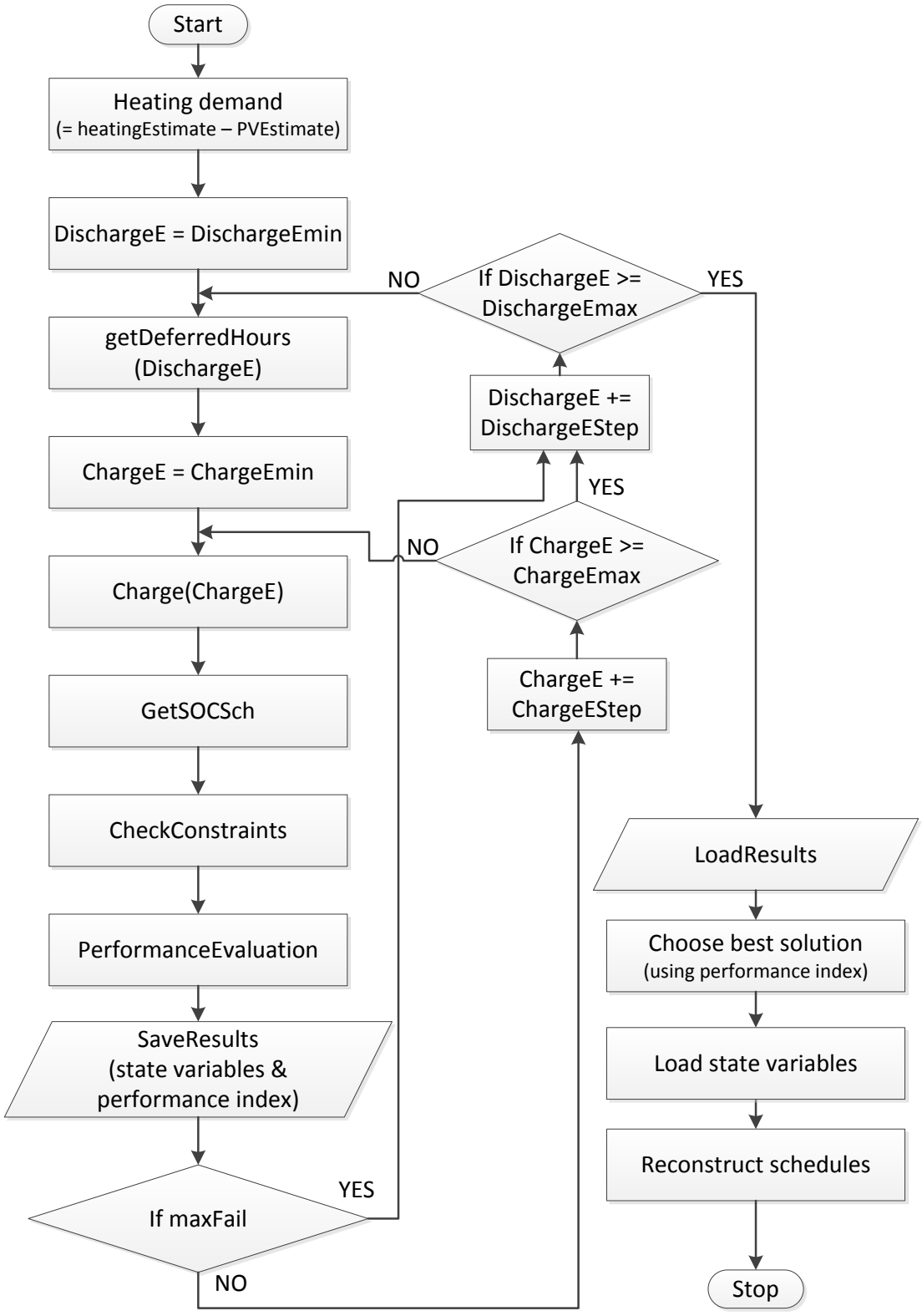


Figure 4.6. *Flowchart of Scheduler*

Secondly, the performance of schedule is evaluated. The performance index is calculated using expenses for purchased energy minus value of energy stored in ESS according (4.7).

$$PerfIdx = purchased - SOC_{kwh}(24) \cdot elspot(24) \qquad (4.7)$$

Where $elspot(24)$ is price of electricity and $SOC_{kwh}(24)$ is state of charge at the end of the day

In last step, values of all state variables with performance index are saved into two lists, one for passing solutions and one for failing ones.

Then, when every state of state-space has assigned performance index with state variables, the best solution can be easily found by taking query with lowest performance index. In case that the list of passing results is empty a failing list is used. When having state variables that produce best schedules, a reconstruction of schedules is done as shown on example in Figure 4.7.

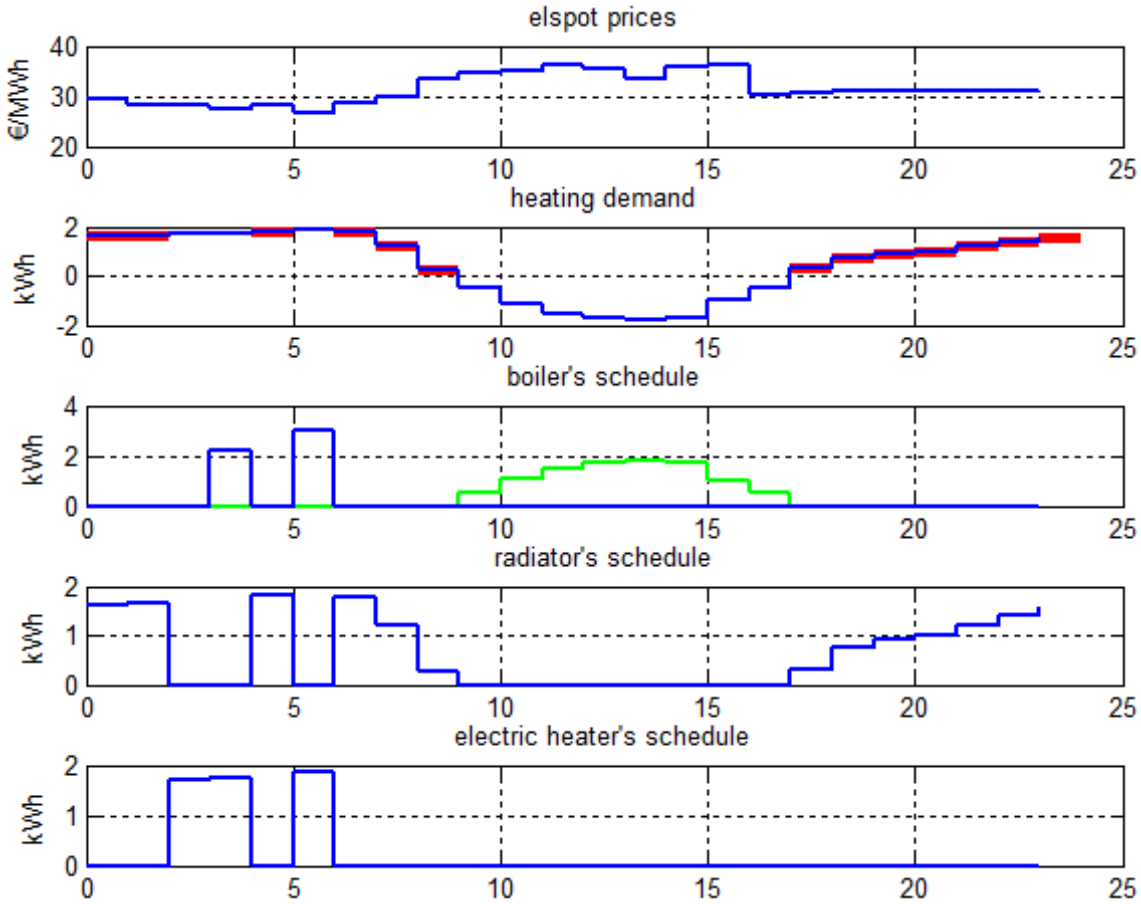


Figure 4.7. *Generated schedules for May 5th 2012*

A proper operation of load shifting is represented by well synchronized operation of boiler and electric heater on elspot prices. As mentioned before, the schedule of radiator is same as heating demand at hours selected for load shifting marked in red colour in Figure 4.7 above. It can be understood, that positive plane of *heating demand* has to be met, therefore sum of *radiator's* and *electric heater's* schedule has to match *heating demand*. Moreover, it can be observed that *boiler's schedule* is charging ESS at times

with lowest price while keeping enough free capacity for local production as can be seen in SOC schedule in Figure 4.5.

Locally produced power that is going to be stored in ESS, depicted in green colour in figure (4.7), is not included in boiler's schedule. It is for the error of weather forecast that results in decrease of algorithm's performance. Charging ESS with locally produced energy is handled by real-time controller that is not affected by error in forecasted data.

This algorithm's weakness is its dependence on precision of irradiance forecast. This becomes an issue when a PV production is much lower than estimated due to e.g. morning fog. The consequence would be lower SOC of ESS than expected and thus it's charging could be forced by *minSOC(40)*, presented in Figures 3.3 and 3.4, in price-peak hours.

To reduce this problem a safety SOC margin (4.6) has been implemented. However a better solution would be having scheduler capable of dynamic intra-day rescheduling. This rescheduling could be done every hour using the most precise forecast available. However, implementation of such scheduler was kept for future research as the complexity of algorithm would be significantly higher.

4.1.3 Weighting scheduler

The algorithm presented in previous chapter does not generate best schedule in every scenario. For that reason another algorithm was developed to work in parallel with the first one. Therefore a combination of both algorithms can result in better performance than a single algorithm could. This chapter explains such scenario together with the algorithm itself.

The scenario that gave reason for this algorithm emerges when hourly prices of electricity have multiple price peaks and PV production is lower than heating estimate. To better understand the benefit of new algorithm an example is provided using data of 15th April 2012.

The algorithm introduced in previous chapter is generating boiler's schedule by selecting cheapest electricity of whole day. It can, however, result in situation when all charging is scheduled to be done in consecutive hours e.g. between 15 and 19 hours as shown in provided example in Figure 4.8 below. This becomes an issue when there are other price-peaks before this period that cannot be fully deferred as the ESS hasn't been scheduled charge beforehand.

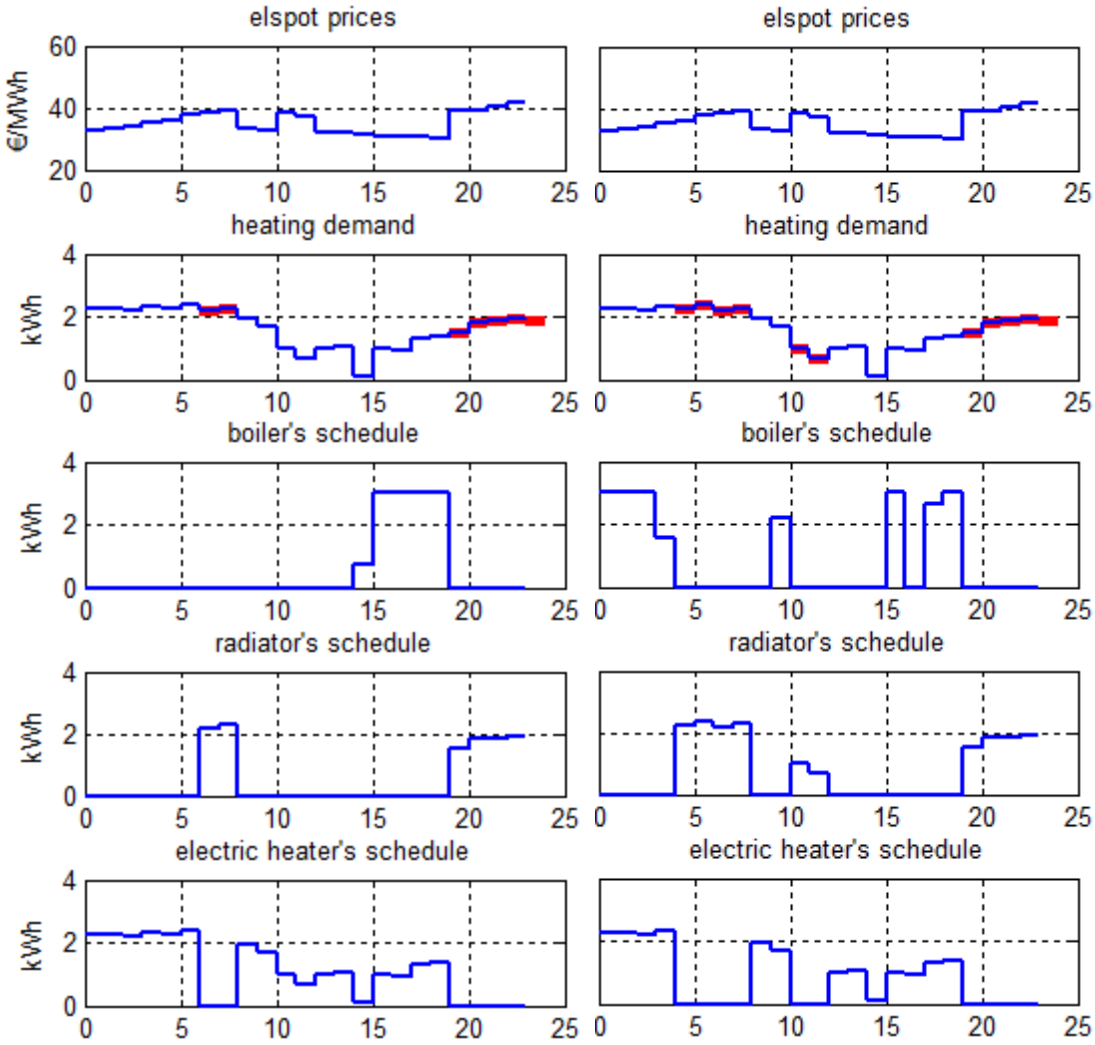


Figure 4.8. *Schedules generated with algorithm 1 and 2*

The best way of understanding this phenomenon is by comparing schedules between the two algorithms shown in Figure 4.8 and between SOC schedules shown in Figure 4.9 and 4.10.

To address this problem, a new strategy of building schedule for boiler has been implemented. A time and amount of energy to be stored into ESS is determined based on hours chosen to be deferred that are marked in red in figure 4.8 above. The radiator's schedule is generated by taking time and magnitude of marked hours. The day is divided into sections according number of deferred sections as can be seen from marked hours for 2nd algorithm in figure 4.8 and dashed lines in figure 4.10. In this example, the day was divided into four sections. Each segment is charged with cheapest available electricity. The amount of energy to be charged within each segment, called *SOCmin*, is calculated as a sum of energy to be deferred within the next section. This energy can be seen as a magnitude of green hyphen at the beginning of each section in Figure 4.10.

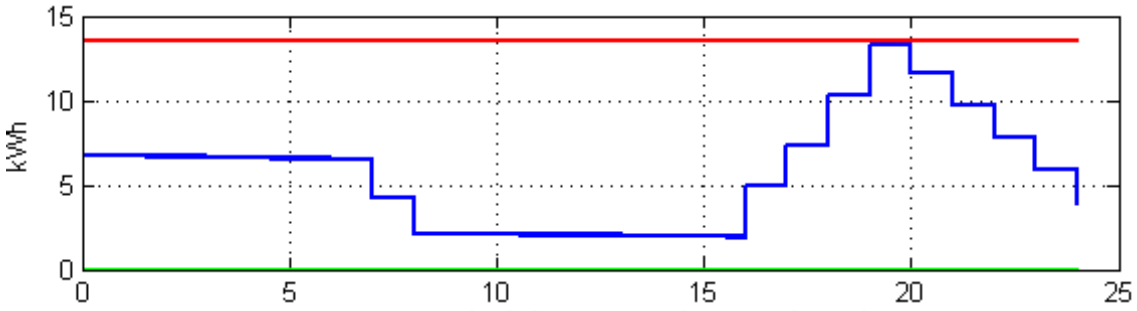


Figure 4.9. *SOC schedule generated using algorithm 1*

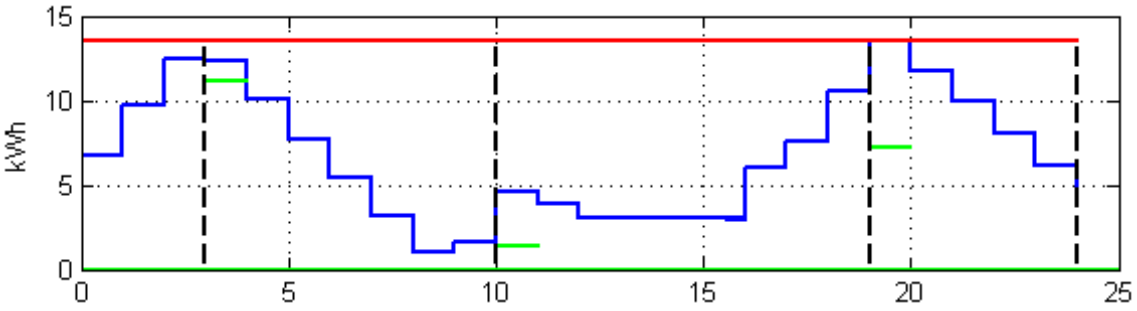


Figure 4.10. *SOC schedule generated using algorithm 2*

The amount of energy to be charged within each segment is decided by energy to be deferred (magnitude of *SOCmin*) and weights as can be seen from formula (4.8) below.

$$E(i) = ChargeE \cdot \big(SegmentCnt \cdot Weights(i)\big) \cdot \frac{SOCmin(i)}{deferredTot} \qquad (4.8)$$

Where $ChargeE$ is one of the state variables described in chapter 4.1.2 that represents energy to be charged in whole day. $SegmentCnt$ is number of sections to be deferred, in this example four, and $deferredTot$ is total energy that is chosen for deferring. It is worth noting, that $deferredTot$ is equal to $DischargeE$, one of the state variables as long as is radiator's rated power sufficient. The vector of weights is considered as third state variable that is determined through iterative process as can be seen from flowchart in Figure 4.11.

This flowchart is an extension of simpler algorithm's flowchart presented in Figure 4.6. A third feed-back loop has been added as the algorithm has now three state variables. This feedback loop goes through "*TuneWeightsRatio*" block depicted on the left-hand side of the flowchart in Figure 4.11. This block decreases weights of all segments where maximum constraints were violated and increases weights of segments where minimum constraints were violated. The weights belonging to segments that didn't result in violation are shifted in order to have unity sum of all weights. In case of violation of maximum or minimum constraints in all segments a further weight tuning would not provide better result. The algorithm therefore continues by changing one of the first two state variables.

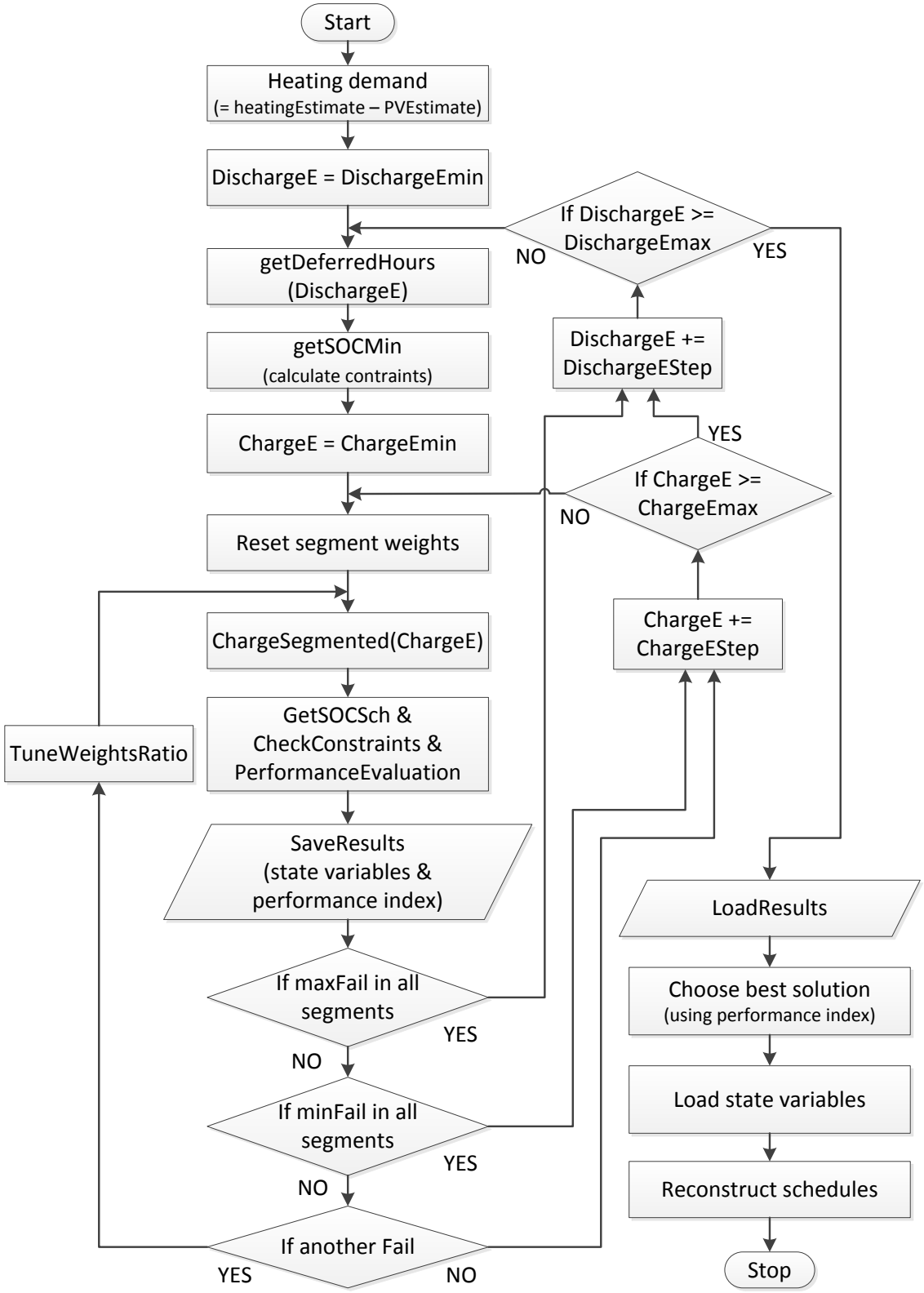


Figure 4.11. *Flowchart of weighting scheduler*

If a maximum SOC constraint is violated in all segments, then a change in weight ratio cannot produce feasible solution and *DischargeE* has to be increased instead. Similarly in a case that a minimum SOC constraint is violated in all segments a *ChargeE* has to be

increased. Therefore, only if another violation emerges, then a ratio of weights is tuned as can be understood from flowchart above.

Simulations showed that weighting scheduler performs 16,5% better than the basic scheduler. The combination of both schedulers increased performance by 23,3%. The achievable monetary savings were therefore increased from 10% to 12,3%. It should be noted that only hourly electricity prices were considered in these simulations. It is because scheduler's algorithm has major influence on load shifting performance and therefore neglecting distribution prices gives better indication of difference between algorithms.

4.2 Real-time controllers

This chapter describes real time controllers that ensure interpretation of schedules, maintaining household's temperature and ensure production following. As can be seen from Figure 4.12, each schedule is interpreted by its own controller and therefore together with production following controller four controllers were implemented.

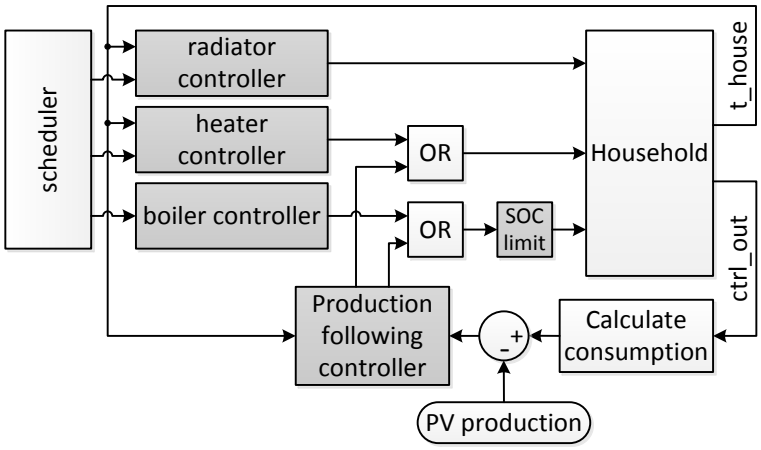


Figure 4.12. *Real-time control scheme*

Boiler controller is the easiest controller to implement as it is only interpreting schedule. The output of this controller is binary and therefore only turn-on and turn-off times have to be decided. In order to minimize energy loss due to thermal insulation of water tank a turn-on signal is generated at the latest of each hour in order to satisfy scheduled energy use. The turn-off time is therefore placed at the end of each hour.

The control signal generated by heater controller cannot be generated with hourly interval as the resulting variations of house's ambient temperature would be too large. Therefore the controller generates pulse-width modulated (PWM) signal with 15 minute period that results in less than 0,5°C variation. The controller is operating in three regimes. First one is for cases when is the household's temperature within specified range from reference temperature and thus the duty ratio of PWM is generated according schedule. The other two are for situations when the house's temperature goes out of specified temperature range. If the temperature goes above specified temperature the control signal is forced low. Similarly, if the temperature is below specified temperature

the heater is forced to run as can be seen from Figure 4.13. The control action is done periodically every 15 minutes that is large enough to prevent feed-back oscillations of closed-loop system and therefore no hysteresis has to be implemented.

The third controller, that generates signal for radiator, is the only controller with continuous output. Its operation isn't much different from heater's controller as can be seen from Figure 4.13 below. A hysteresis highlighted in grey colour had to be implemented as this controller is operating at the same sampling rate as the rest of the simulation and therefore the oscillations need to be prevented.

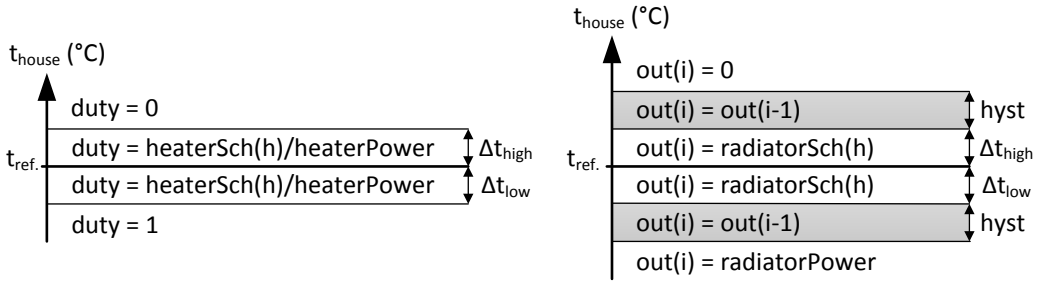


Figure 4.13. *Heater's and radiator's control*

The hysteresis can be understood as transient area in which no change is applied. The expression in Figure above is using two different time variables, where *i* represents simulation time, that increments every simulation step, and *h* represents an hour that is needed to specify query in schedule to be used.

The parameters of radiator's controller have to be stricter then heater's in order to minimize electricity purchase beyond scheduled volumes. The parameters chosen for these controllers are as follows.

$$\begin{aligned} \Delta t_{heater\_high} &= 0.0°C \\ \Delta t_{heater\_low} &= 0.1°C \\ \Delta t_{radiator\_high} &= 0.6°C \\ \Delta t_{radiator\_low} &= 0.6°C \\ hyst_{radiator} &= 0.1°C \end{aligned} \tag{4.9}$$

The last controller is called *Production following controller*, as it is responsible to respond to local PV production by controlling electric heater and boiler as depicted in Figure 4.12.

The main goal of this algorithm is to minimize energy injected to the grid. This strategy is, however, not the best strategy in every scenario. It is mainly in summer when storing excessive production into heat has no justification as there is no expected use for it. This is the reason why a *SOClimit* had to be implemented, that will be explained later.

The controller inputs net consumption of household that is expressed as a difference of house consumption and local PV production. This input is being integrated with hourly reset and as long as the integrated value is positive, no driving signals are gener-

ated. Both loads controlled by this controller are binary and therefore pulses need to be generated. Turn-on times of loads are decided by the time required for the integrator to return to zero. The load is switched-on if the condition (4.10) is satisfied.

$$\begin{aligned} E_{int} &< -E_{load} \\ sum \cdot Ts &< -P_{load} \cdot T_{onMax} \end{aligned} \tag{4.10}$$

Where sum is the integrated input power that is multiplied with simulation step size Ts in order to get energy in Watt-seconds. The simulation step size Ts has been set to 20 seconds as this value has been found to be the largest without effect on simulation results. The T_{onMax} is maximum on-time of generated pulse that is set to 15 minutes. And the P_{load} is load's rated power that is substituted by power of one of the loads according to household's temperature. The main part of this controller is implemented as state machine as shown in Table 4.1 below.

Table 4.1 *Table of production following algorithm*

	$\boldsymbol{E_{int} \geq 0}$	$\boldsymbol{0 > E_{int} \geq E_{load}}$	$\boldsymbol{E_{load} > E_{int}}$
$T_{house} \geq T_{ref}$	$boiler = 0$ $heater = 0$	$boiler = no\ change$ $heater = 0$	$boiler = 1$ $heater = 0$
$T_{house} < T_{ref}$	$boiler = 0$ $heater = 0$	$boiler = no\ change$ $heater = no\ change$	$boiler = 1$ $heater = 1$

Where *no change* keeps the previous state and therefore the range between zero and E_{load} is a hysteresis band. However, there is exception for heater signal that goes low as soon as the temperature of house rises above reference temperature T_{ref}.

Figure 4.14 below show the operation of controllers. The simulation has been done for 3rd July 2012 and only mid-day is shown as we are interested in situation when the local production exceeds local consumption. An integrated energy E_{int} is shown in first graph. It returns towards zero by the end of each hour due to generated signals shown in 3rd and 4th graph. The second graph shows difference between ambient and reference temperature. The photovoltaic production is captured in 5th graphs and the resulting power flow at the mains is depicted in last graph. This power is never exactly zero as the perfect production following isn't achievable with discrete loads. However, its average is zero as represented by E_{int} being close to zero at the end of each hour.

It is important to realize that, in summer months, the heating requirement is very small or even zero. In this situation, there is no need to fully charge water boiler as it is not likely to utilize such a big amount of heating energy. It should be noted that in this case-study a water tank is used only for space heating and thus its other use is not considered.

This is the reason why a *SOCLimit* had to be placed in front of model of the Household on the path of boiler's control signal as depicted in Figure 4.12. This system is, as its name suggests, limiting boiler's state of charge.

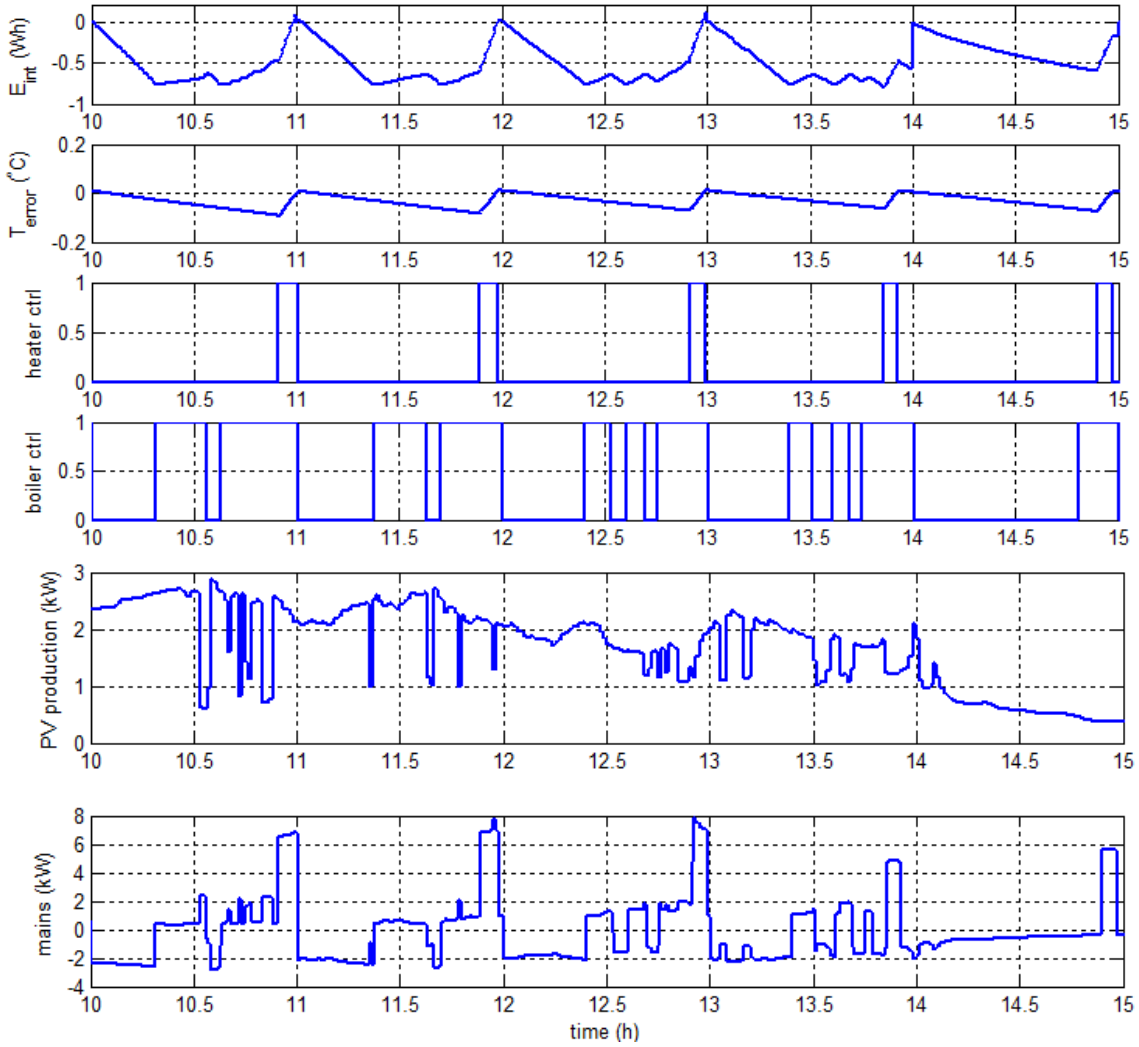


Figure 4.14. *Output of production following controller*

The best indication of future use of stored energy is a *HeatingEstimate* calculated from weather forecast, required indoor temperature and thermal insulation that has already been expressed for scheduler as discussed in chapter 4.1.1. Having this estimate, an implementation of this system is done using expression (4.11)

$$
\begin{aligned}
&if\ (SOC\ > 1.3 \cdot HeatingEstimate)\{\\
&\qquad ctrl_{out} = 0;\\
&\}else\{\\
&\qquad ctrl_{out} = ctrl_{in};\\
&\}
\end{aligned}
\tag{4.11}
$$

Where SOC is state of charge of water boiler in kilowatt-hours and 1,3 is a limitation factor that has been set based on set of simulations that are presented in chapter 5. This figure is unique to parameters of house, water boiler and heaters used for this case study.

5. SIMULATIONS

All simulations that have been done are categorized into three chapters. First one is focusing on effect of household setup like size of PV installation and energy storage system. Second one is giving an insight into effect of weather forecast error. The last set of simulations had to be done in order to configure parameters of the algorithm itself.

All simulations are presented as a function of percentual savings compared to household without any automation. Simulations were run using data of year 2012 provided by meteorological station at Tampere University of Technology [18]. Every simulation has been run for whole year with simulation step size set to Ts = 20s in order to get realistic representation of algorithm's performance.

In order to express monetary savings and thus express performance of algorithm a *Cost calculator* had to be implemented.

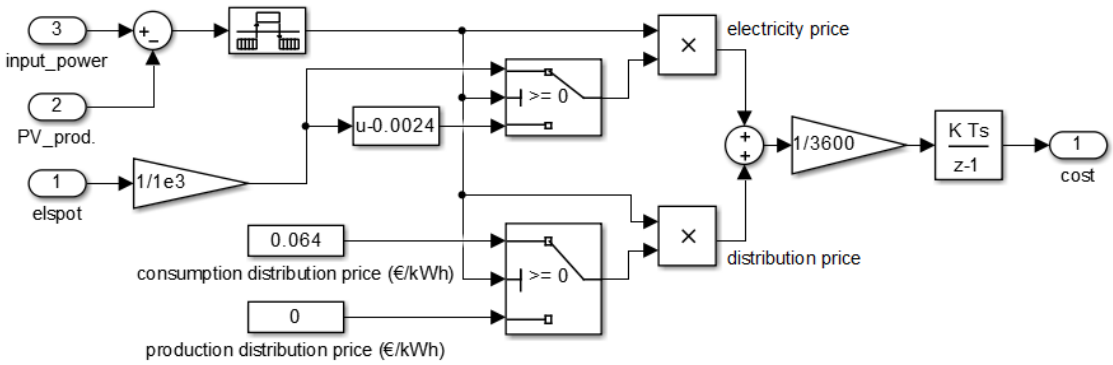


Figure 5.1 *Cost calculator*

The figure 5.1 above is representing implementation of such calculator. It has three inputs. An *input_power* is providing current consumption of all appliances, *PV_prod.* is providing produced power by PV panels and *elspot* is providing hourly prices of electricity provided by Nordpool day-ahead market in €/MWh [20]. The total cost is composed of two parts, an electricity cost and distribution cost. Both of these have different pricing for selling and purchasing. The price of electricity uses hourly pricing provided by Nordpool. A tariff by Fortum gives customer 0,24c/kWh less than current hourly price [21]. The distribution price is 6,4c/kWh for purchased electricity and 0c/kWh for sold electricity. There can, however, be a tariff where a customer would pay up-to 0,07c/kWh for distribution of produced electricity.

This calculator is implemented twice, once measuring mains of the house and once measuring its power use. The power used by the house is heat power that is converted into equivalent of electrical power. It should be emphasized that simulated model

is simplified version of realistic household and therefore only heating loads are considered.

The default error introduced into input data, as presented in chapter 4.1.1, is ±3°C for temperature and ±25% for irradiance. The default size of PV installation is 4kWp, size of water tank is 290litters and the total transmittance of house is 100W/K.

The implementation of load shifting results in 4,2% savings that, for a simulated household, translates to annual savings of 57€. When a production following controller is enabled, then savings is increased to 5,6% giving annual savings of 77€. The performance throughout the year is depicted in Figure 5.2 below. It can be seen that most savings is achieved in winter months peaking with 19€ in February. On the contrary the maximum percentual savings is achieved in middle of the summer with 30%. These summer savings are a product of production following controller described in chapter 4.2.

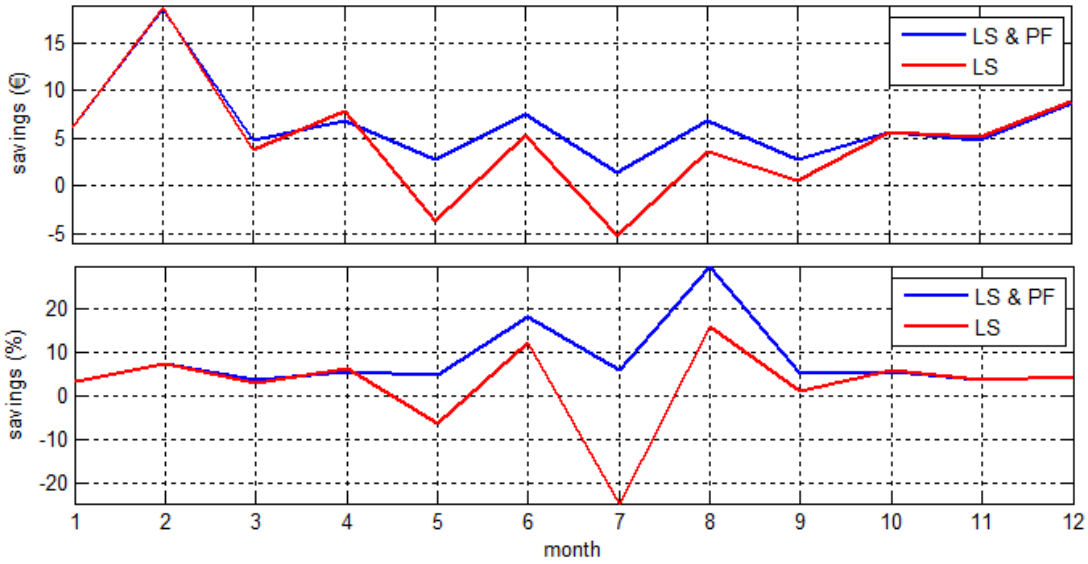


Figure 5.2 *Monthly savings*

5.1 Effect of household setup

The size of energy storage system (ESS) is crucial when a meaningful savings are desired. A Figure 5.3 shows effect of ESS size on savings. First graph is scaled to volume in litters. The second graph is scaled to capacity in kilowatt-hours to give better idea of system size when considering e.g. battery storage alternative. This conversion assumes that the temperature of hot water boiler stays between 40 and 80 degrees Celsius.

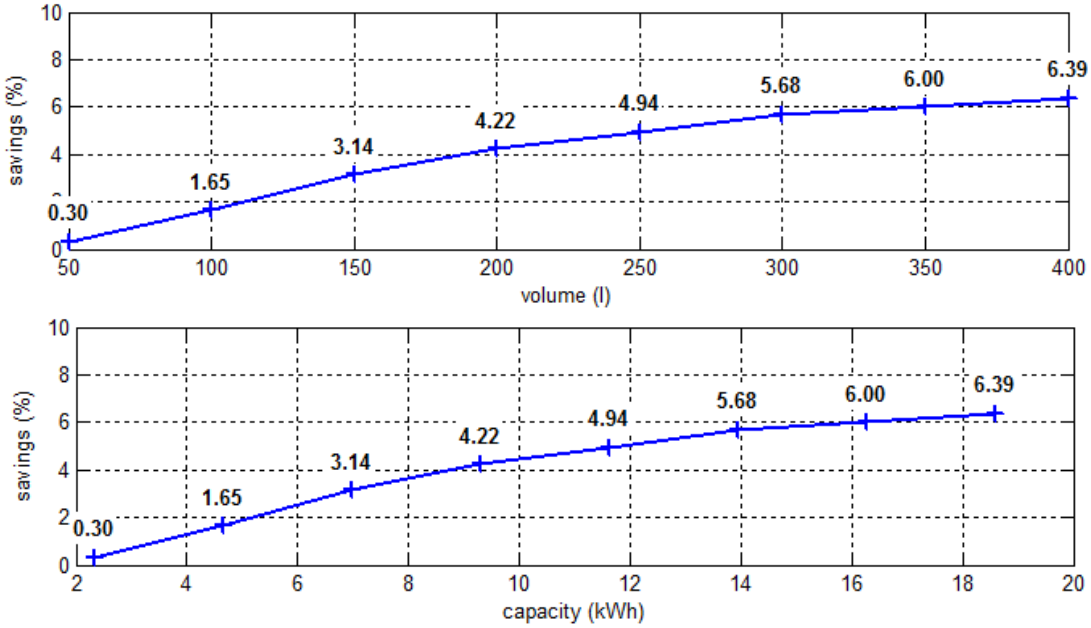


Figure 5.3 *Effect of ESS size on performance*

A simple analysis has been done in order to determine the best size of ESS. The total cost of the system would comprise of constant price for controllers, contactors, etc. and variable price according size of ESS. The analysis represented by table 5.1 below is considering the price only for ESS. The fastest return of investments can be achieved with 200 litter unit. However, if its lifetime is assumed to be 20 years, then a 300 litter alternative results in higher savings.

Table 5.1 *Effect of water tank size on payback and revenue*

Volume (l)	***ESS investment (€)***	***Annual savings (€)***	***Payback (years)***	***Lifetime savings (€)***
100	*176*	*3,5*	*50,2*	*-105,7*
150	*219*	*29,5*	*7,4*	*371,2*
200	*269*	*51,5*	*5,2*	*761,4*
250	*416*	*65,5*	*6,4*	*894,0*
300	*549*	*79,5*	*6,9*	*1041,3*

Total savings is represented by the last column called ”*Lifetime savings (€)*”. If we assume that the constant price of system is 200€, then the overall savings would be 841€.

The next decision of a customer would be about size of PV system. The simulations, presented by Figure 5.4 below, show that the savings potential increases with the size of PV installation. It is because production can match consumption more often resulting in longer operation time of production following controller.

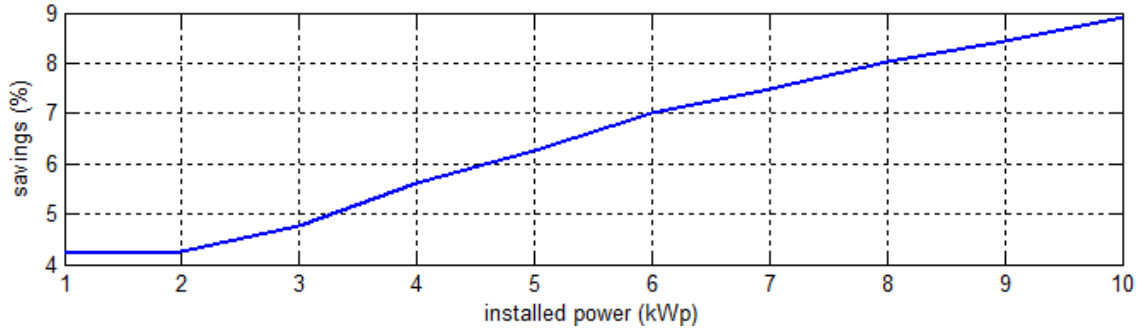


Figure 5.4 *Effect of PV system size on performance*

The last important parameter affecting savings is transmittance. Transmittance is affected by the surface area of outer walls and its insulation. In order to simulate houses with bigger values of transmittance a power rating of heaters had to be increased so required room temperature could be maintained. Simulation results, presented in Figure 5.5, are showing decreasing performance with increasing transmittance. The reason for that is constant capacity of energy storage system. Increasing power flows within household without corresponding increase of ESS capacity decreases potential of load shifting and thus achievable savings.

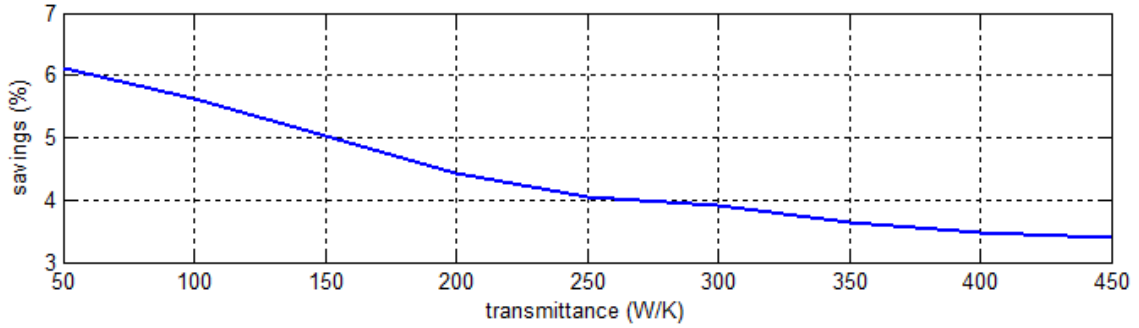


Figure 5.5 *Effect of house transmittance on performance*

5.2 Effect of input data error

An error has been introduced into outdoor temperature to represent error of weather forecast. This forecast data is used by scheduler as described in chapter 4.1.1. The efficiency of algorithm is very little dependent on introduced error as can be observed from Figure 5.6 below.

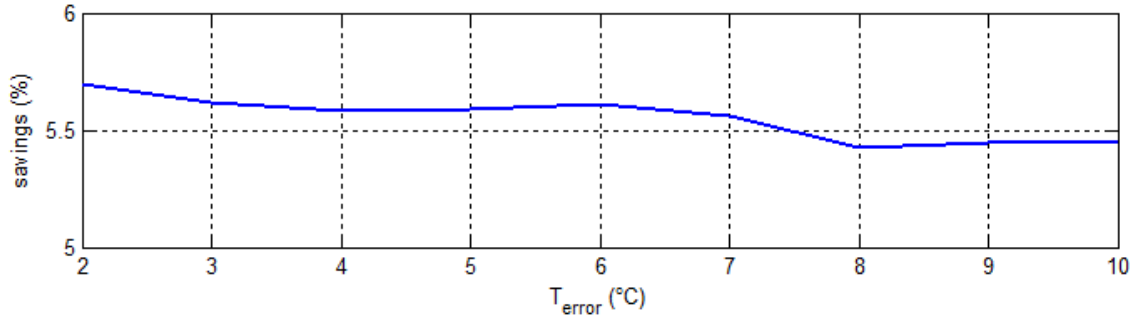


Figure 5.6 *Effect of temperature forecast error on performance*

The error of irradiance forecast has very small effect on savings as shown in Figure 5.7 below. It is because a production following is implemented as a real-time controller. The small decrease in performance is due to scheduler using irradiance forecast to determine reserve in ESS for photovoltaic production.

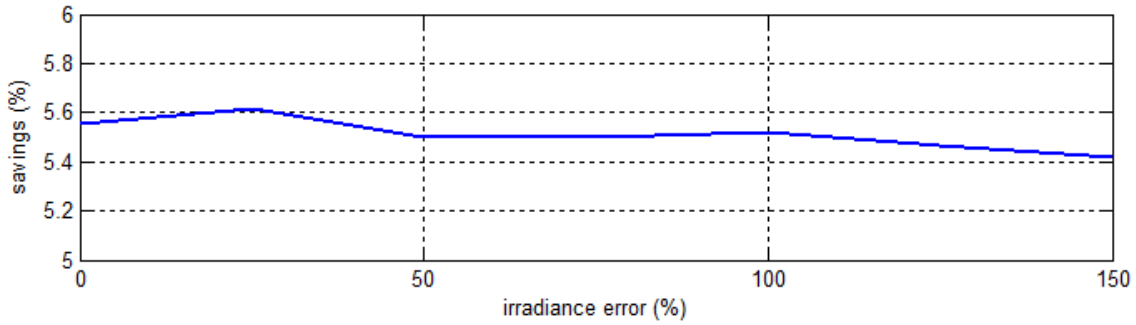


Figure 5.7 *Effect of irradiance forecast error on performance*

5.3 Effect of algorithm settings

This chapter presents simulations that have been done in order to determine optimal settings of algorithm. These settings concern scheduler that affects load shifting performance. This is the reason why only hourly prices of electricity were considered when evaluating algorithm's performance. This is why the presented savings are different from savings presented in previous chapters.

First parameter that has been simulated is *SocEndMax*. This parameter is used by scheduler as a constraint for state of chare of ESS at the end of the day. It is, as discussed in chapter 4.1.2, needed to enable purchase of cheap electricity in the first hours of upcoming day. Figure 5.8 suggests that the best coefficient for expression 4.5 is equal to 0,55.

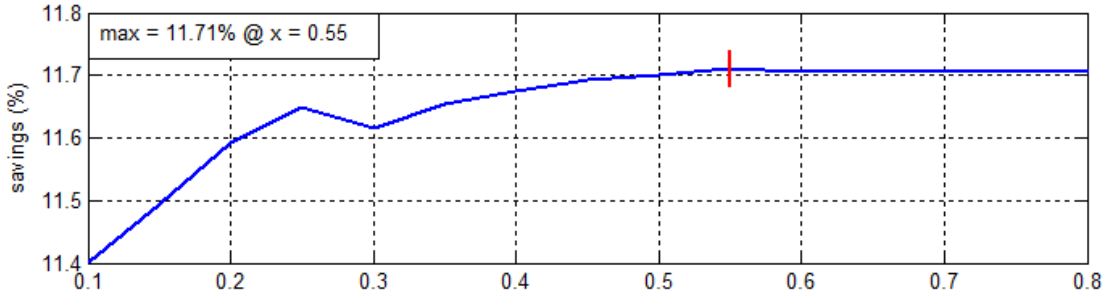


Figure 5.8 *Effect of "SOCEndMax" on performance*

Another two parameters are safety margins of SOC. These are needed to compensate for error in input data. Figures 5.9 and 5.10 are showing that the best parameters for equations in 4.6 are 0,01 for *SocMinMargin* and 0,02 for *SocMaxMargin.*

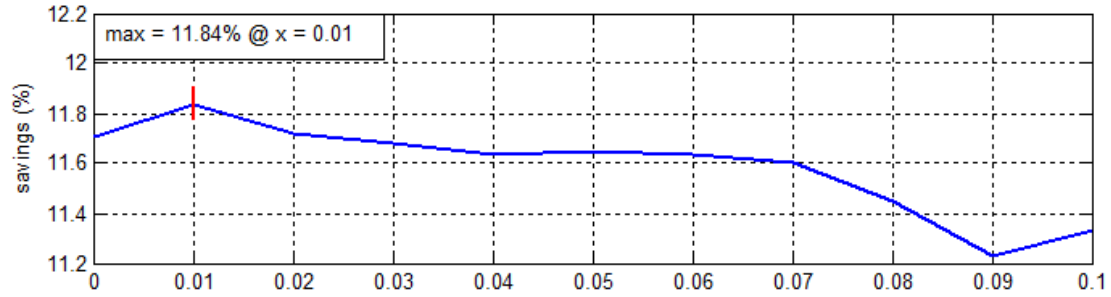


Figure 5.9 *Effect of "SOCMinMargin" on performance*

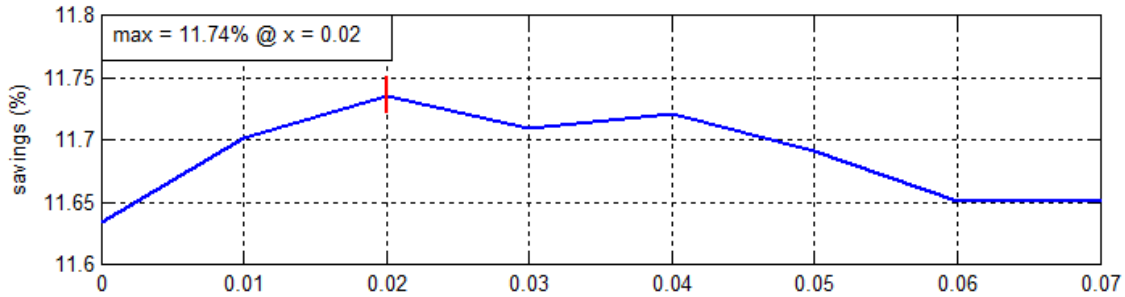


Figure 5.10 *Effect of "SOCMaxMargin" on performance*

Next parameter called *noOfSteps* affects the number of values assigned to two state-variables of scheduler as explained in chapter 4.1.2. Its value is therefore affecting processing time and memory footprint. The scheduler is generated only once a day and therefore only memory requirement is considered. The simulation result, presented in Figure 5.11, is showing both, percentual savings and memory footprint. A value of 36 steps results in scheduler using reasonable 61kB of RAM. Having this small memory footprint enables use of cheap and low-power M class ARM processor like STM32F4.

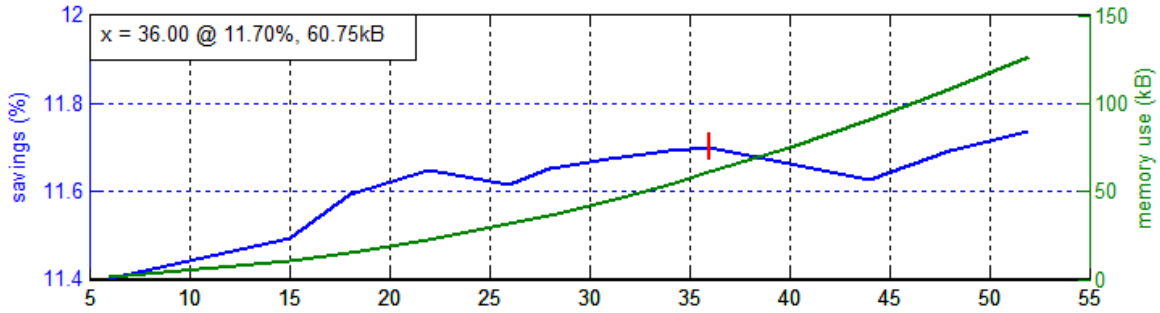


Figure 5.11 *Effect of "noOfSteps" on performance*

Another parameter called "*SOCLimit"* is part of real-time controller. It limits maximum SOC of ESS as introduced in chapter 4.2. Simulation results, presented in Figure 5.12, clearly show that the best parameter for equation 4.11 is equal to 1,3.

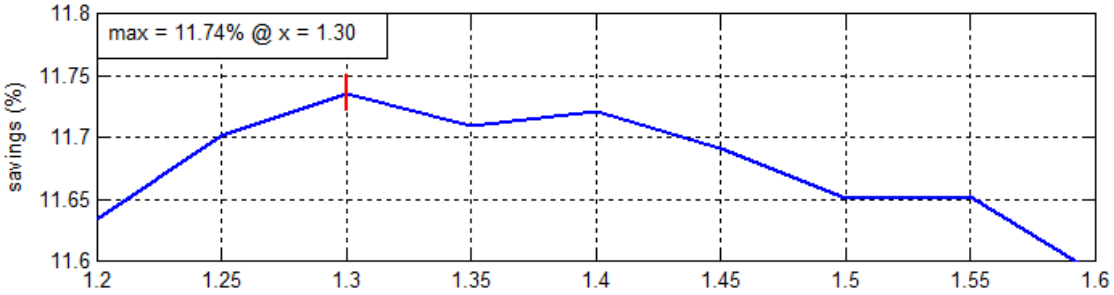


Figure 5.12 *Effect of "SOCLimit" on performance*

6. DEVELOPMENT OF AC MICROGRID LABORATORY

This chapter describes development of AC Microgrid laboratory. The design was following initial idea shown in Figure 2.1 where a dSpace and HEM PC can be seen. These two units are core elements of laboratory as they manage behaviour of whole. First, a dSpace unit is introduced in chapter 6.1, then its use for load switching is explained in chapter 6.2 and finally a data acquisition is described in chapter 6.3. The most important measurement is power that is needed by HEM PC and its production following algorithm that has been described in chapter 4. In order to provide these power readings to HEM PC a digital link, that is depicted in Figure 6.1, has been implemented that is described in chapter 6.4. Then, a HEM PC and its implementation of Z-Wave wireless technology is described in chapter 6.5. And finally the last chapter demonstrates basic functionality of developed laboratory.

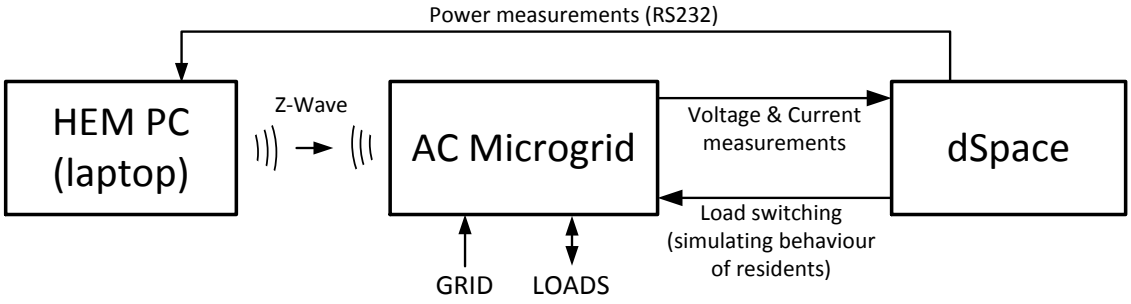


Figure 6.1. *AC Microgrid laboratory schematic*

The physical implementation of AC Microgrid has been built according schematic captured in Figure 2.1 that resulted in system depicted in Figure 6.2 and 6.3. It has two 3phase contactors in series for each of seven channels. The first row of contactors is controlled by HEM PC using wireless standard designed for home automation called Z-Wave. The second row is controlled by dSpace using solid state relays (SSRs) to emulate e.g. resident's behavior. This work comprised of design and implementation of power measurement, control of SSRs and implementation of wireless Z-Wave relays. The current is sensed using current transformers (CTs) and the voltage is sensed by voltage transducer at bus-bar. All these parts are marked in colour in Figures below and will be described in following chapters.

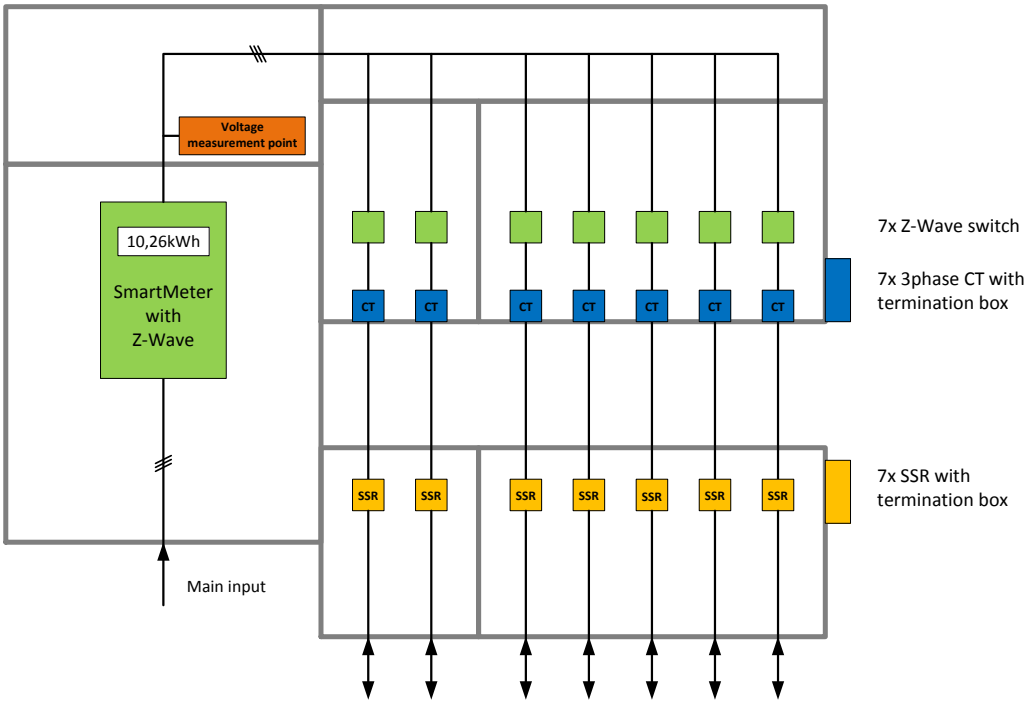


Figure 6.2. *AC Microgrid schematic*

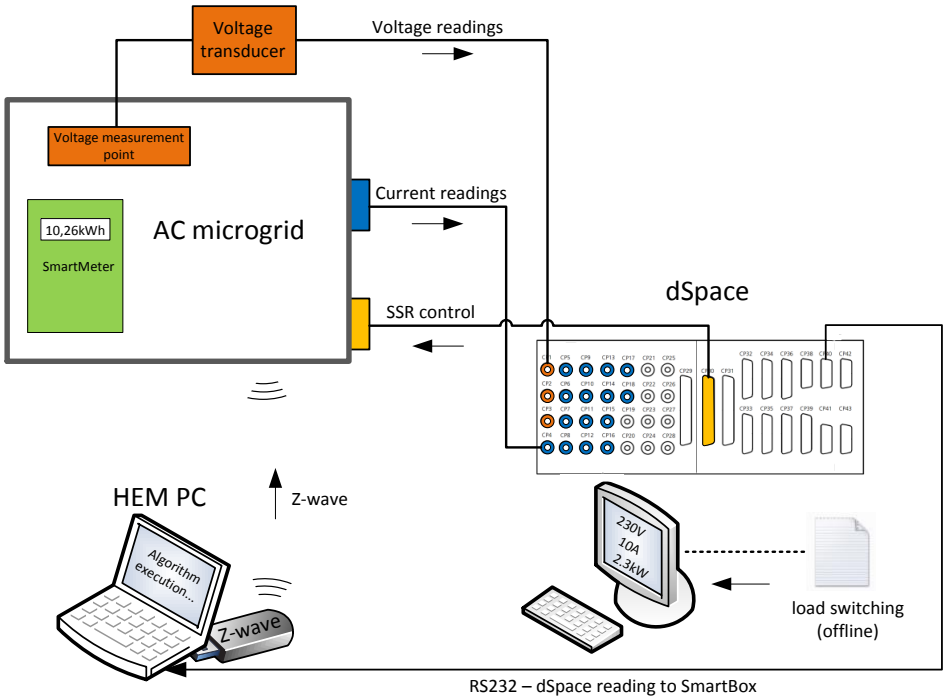


Figure 6.3. *AC Microgrid schematic*

6.1 dSpace

A dSpace is a German company that specializes in rapid prototyping solutions that can dramatically ease development process by allowing Simulink models to interface with real-word peripherals in real-time. The dSpace model used in this project is DS1103 having 20 analog inputs, 8 analog output, 32 universal digital inputs/outputs, RS232/RS485 and CAN serial interface [22].

AC Microgrid has 7 three phase channels that could utilize 21 analog inputs for current measurements plus three for voltage measurement. Therefore 24 channels could be used. It is, however, sufficient to measure five channels at once for the first phase of the project. Therefore, only 18 analog inputs will be used keeping two inputs free for possible future upgrade – e.g. temperature measurements. Not all analog to digital converters in dSpace are of the same type. They differ by resolution, sampling speed and multiplexing as shown in Table 6.1.

Table 6.1 *Specification of dSpace's A/D converters*

	ADC1 – ADC4	***ADC5 – ADC8***
multiplexed inputs	*Yes*	*No*
number of inputs	*16 (4x4)*	*4*
input span	*±10V*	*±10V*
sampling frequency	*250kHz*	*1,25MHz*
resolution	*16bit (0,3mV)*	*12bit (4,9mV)*
offset error	*± 5mV*	*± 5mV*

The bandwidth of converters couldn't be fully utilized because the digital signal processor (DSP) within dSpace cannot process more than 5000 samples per second (sps) in real-time. The number of active channels has significant influence on maximum possible sampling speed as shown in Table 6.2 below. The slower ADCs with 250kHz sampling frequency are sensing four inputs due to input multiplexer. The multiplexer itself takes 2µs to switch [23]. Therefore approximately 55,5ksps sampling can be achieved.

Table 6.2 *Sampling speed according number of acquisition channels*

number of channels	*1*	*2*	*3*	*4*	*5*
sampling speed (ksps)	*5*	*4*	*3*	*2*	*2*

Throughout the development process the Simulink model was kept with all five channels and therefore its settings is set to 1/2000 second Fixed-step simulation size with discrete solver. Digital outputs of dSpace are capable of delivering up to 10mA that, as is concluded in chapter 6.2.1, is enough to drive solid state relays (SSR) directly without any amplification.

6.2 Load switching – resident's emulation

This chapter describes an implementation of load switching by dSpace. The implementation can be divided into three parts, each having its own chapter. First, an electrical interface is explained in chapter 6.2.1. Then a data representation suitable for this application is introduced in chapter 6.2.2 and finally a Simulink model that generates the driving signal is explained in chapter 6.2.3.

6.2.1 Interface between dSpace and contactors

Datasheet of dSpace defines output voltage to be 5V with 33Ω resistor in series with maximum output current of 10mA [23]. The minimum input voltage of solid state relay is said to be 4V at which the drawn current is 13.5mA [24]. According these specifications an amplification of driving signal seems necessary. A presence of amplifier would require external power supply that would complicate design and therefore an alternative without need for amplifier was researched more thoroughly. First, the minimum input voltage and current at which a SSR reliably switches was measured. Besides that, a voltage on terminals of SSRs was measured at 10mA as shown in (6.1).

$$\begin{aligned} V_{SSR\_min} &= 2{,}52V \\ I_{SSR\_min} &= 5{,}27mA \\ V_{SSR\_10mA} &= 3{,}11V \end{aligned} \tag{6.1}$$

Then voltage at the output of dSpace at its maximum current and at minimum current of SSR has been measured.

$$\begin{aligned} V_{dSpace\_10mA} &= 3{,}17V \\ V_{dSpace\_5{,}3mA} &= 3{,}48V \end{aligned} \tag{6.2}$$

Having these measurements, it can be concluded, that dSpace can drive SSRs directly without any amplifier and only a series resistor needs to be added, where its minimum and maximum values are calculated according (6.3) below.

$$\begin{aligned} R_{min} &= \frac{\mathrm{V_{dSpace\_10mA}} - \mathrm{V_{SSR\_10mA}}}{10mA} = 6\Omega \\ R_{max} &= \frac{\mathrm{V_{dSpace\_5.3mA}} - \mathrm{V_{SSR\_min}}}{\mathrm{I_{SSR\_min}}} = 181\Omega \end{aligned} \tag{6.3}$$

Therefore the average value of 100Ω has been chosen and integrated into terminal box that is attached on the right side of AC Microgrid as can be seen on photo of prototype in Appendix 1.

6.2.2 Representation of driving signal

Driving signal controlling contactors originates in data file in CSV format. Data of this signal is downloaded to dSpace together with the machine code after Simulink model compilation. For this reason a memory efficient representation of data is preferred. Beside that any unnecessary data conversion should be avoided as dSpace has limited processing power. Therefore an asynchronous binary format has been used with data queries present only on times of signal changes. An example of such data representation is shown in Table 6.3 below.

Table 6.3 *Driving signal representation example*

time (s)	*data (base 10)*	*data (base 2)*	*switched SSRs*
1	*2*	*00000010*	2^{nd}
2	*4*	*00000100*	3^{rd}
4	*6*	*00000110*	2^{nd} & 3^{rd}

Thanks to asynchronous data representation a duplicate queries could be omitted as can be seen on provided example for third second. The data itself, when written in binary format, directly represents state of SSR switches when read from right as can be seen from last two columns in table above.

6.2.3 Simulink model

A Simulink model that outputs binary data is thanks to efficient data representation quite simple as can be seen from Figure 6.4. A block *From Workspace* is used to load data. *Holding final value* in its setting has to be chosen in order to disable approximation between consecutive values. Then data conversion block had to be used, that doesn't require any extra processing time as it only affects compilation. Finally, driving signals are outputted using proprietary *DS1103BIT_OUT8_GO* block that is part of dSpace library.

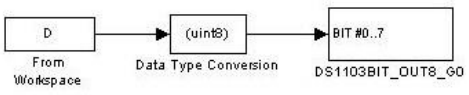


Figure 6.4 *SSR control interface box*

6.3 Measurement system

Production following algorithm needs real-time readings of power as discussed in chapter 4.2. The Smart Meter couldn't be used as it doesn't provide real-time readings. To measure power, a voltage and current needs to be sensed. Therefore a voltage and current transducers had to be developed that are described in chapters 6.3.1 and 6.3.2 respectively. Then, an algorithm processing raw data of ADCs is described in chapter 6.3.3.

6.3.1 Voltage transducer

Voltage transducer is a device that attenuates voltage to amplitude compatible with dSpace. The transducer is dealing with life-threatening voltage levels and therefore a safety had first priority in design process.

A step-down 230V/6V 1W transformer has been used to achieve galvanic isolation that achieves satisfactory safety and eliminates ground loop current issue. The selected transformer wasn't best choice as the specified output voltage is achieved only when loaded with nominal power. Therefore, in order to attenuate transformer's output voltage a voltage divider has been added. The divider is composed of two 100Ω resistors that divides the voltage in half and acts as 200Ω load. This setup results in 1:60 attenuation that gives about $10{,}83V_{pp}$ output when measuring $230V_{RMS}$. It is well within ADCs specifications as the maximum input voltage is $20V_{pp}$ [23]. When connecting transducers outputs to 16bit ADCs a theoretical resolution is about $13mV_{RMS}$.

The transformer is protected from short circuit current with resettable fuse rated to 100mA. A voltage spike could be induced into cable between transducer and dSpace and therefore a transient voltage suppressor (TVS) has been added to protect inputs of dSpace. A one phase schematic of such transducer is shown in Figure 6.5 below.

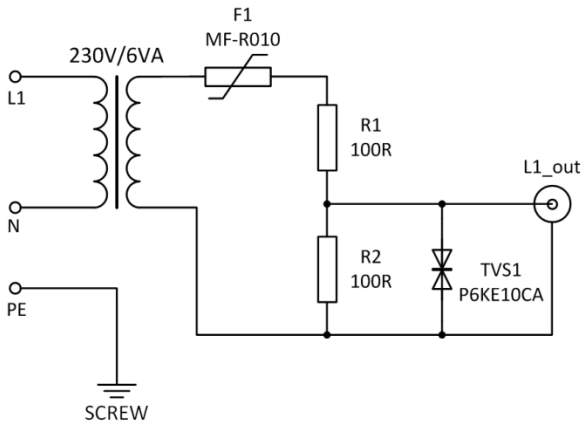


Figure 6.5. *Schematic of 1phase voltage transducer*

The fact that the transducer is installed in plastic box and the use of isolation transformer resulted to no use of protective earth (PE) wire. It was, however, terminated by screw at the side of transducer to allow its grounding with the rest of the system. Three-phase voltage transducer has been built by placing three one phase transducers independent of each other in one box as can be seen on photos of prototype in Appendix 1. Attenuated signals are accessible via three BNC connectors each providing signal of one phase. However, to minimize the number of cables used, a 6pin connector has been added that comprises signals of all phases. Cable used for this three phase signal comprises of three twisted pairs that are shielded so measurements shouldn't get affected even in environment with high electromagnetic noise.

6.3.2 Current transducer

The safety and simplicity of designed solution was of high priority. The three main non-intrusive sensing methods are hall-effect based sensors, Rogowski coils and current transformers that by its nature eliminate any safety constraints as they are not electrically connected to conductor of measured current [25]. First two approaches include electronics that requires external power that would increase complexity and therefore negatively affect reliability. Therefore a current transformer has been chosen, even-though it is not open loop and cannot be clapped on measured wire. Instead, it has to be permanently installed into the system. Besides that, the required bandwidth is rather small and therefore using current transformer not capable of measuring DC and high frequencies is of no issue. Last, but not least, its dimension is small enough to fit into the AC Microgrid panel nicely as can be seen on photo of prototype in Appendix 1. In case, that wide-bandwidth measurement would be required, an additional clamp-on probe can be installed on freely accessible phase wires on AC Microgird panel.

A current transformer by Amveco - AMC-1020 with $20A_{RMS}$ nominal and $60A_{RMS}$ maximum range has been chosen. A burden resistor is required in order to convert output current of transformer into voltage for which a recommended value of 100Ω has been used. A burden resistor of this value results in $2{,}83V_{pp}$ output when sensing $20A_{RMS}$ that is with good margin within range of dSpace. Schematics of transducer can be seen in Figure 6.6 below. The output of three current transducers representing one channel is interfaced with dSpace by the same cable as the voltage transducer that is a shielded cable with three twisted pairs. When a current transducer is connected to 16bit or 12bit ADC the theoretical resolution is $2.2mA_{RMS}$ and $34.5mA_{RMS}$, respectively.

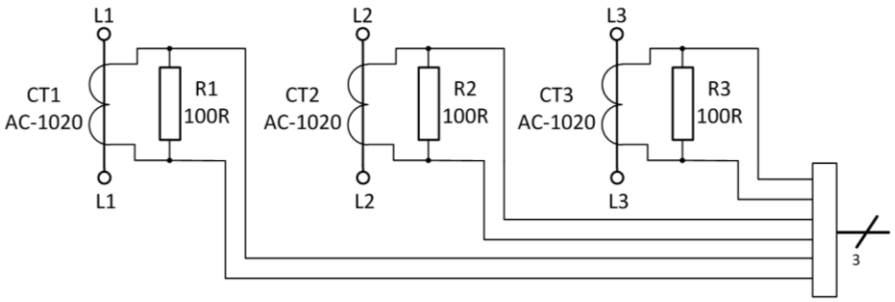


Figure 6.6. *Schematics of 3 phase current transducer*

6.3.3 Data acquisition model

A Simulink model that processes data stream of ADCs sensing voltages and currents provides variety of readings that are listed in Table 6.4 below. The sampling speed of ADCs was set to 1/2000 of second that corresponds to 500Hz. It means, that the sampling frequency is 10x higher than the frequency of measured grid frequency and therefore Nyquist-Shannon sampling theorem was satisfied [26].

Table 6.4 *Measured parameters*

Name	***Symbol***
effective current	I_{RMS} *(A)*
effective voltage	U_{RMS} *(V)*
frequency	*f (Hz)*
power factor	*PF (-)*
apparent power	*S (VA)*
active power	*P (W)*
reactive power	*Q (VAr)*
apparent energy	E_S *(VAh)*
active energy	E_P *(Wh)*
reactive energy	E_Q *(VArh)*

The model is structured using subsystems where the top layer comprises of five *3phaseMeters* representing five channels of AC Microgrid, common acquisition subsystem, subsystem responsible for calculating effective value of voltages and calibration subsystem as can be seen from Figure 6.8.

A common acquisition is encapsulated in *CommonAcq* subsystem that provides three outputs that are for simplicity of wiring multiplexed into one bus as can be seen in Figure 6.7. First output is a signal at half of grid's frequency that provides triggering for phase-detect blocks and resetting signal for integrators of RMS calculators. The second output is used for filtering and limiting update rate of output readings. The last, third, output provides measured frequency that is used by effective value calculators and phase-detect blocks.

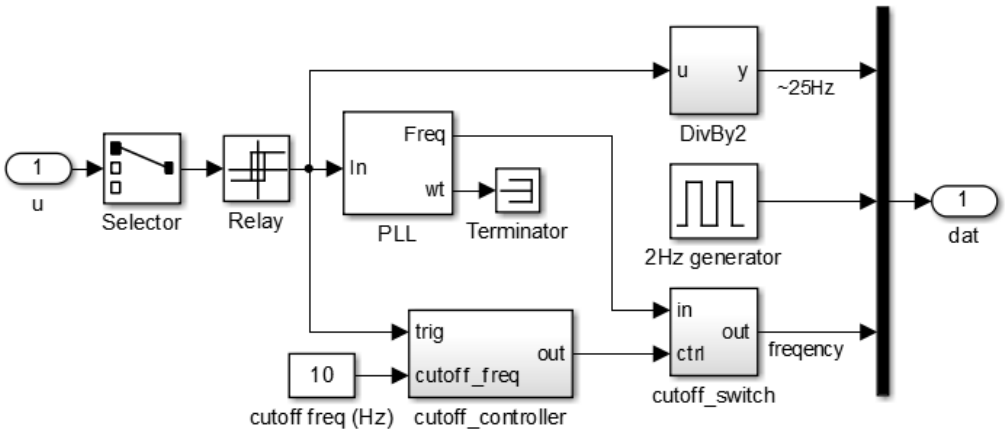


Figure 6.7 *Common acquisition subsystem*

The frequency is measured using PLL block that gives precise readings but isn't capable of detection zero frequency in case of loss of mains. Therefore in order to avoid misreading a *cuttof_controller* has been added that in case of smaller that 10Hz frequency forces the output frequency to zero.

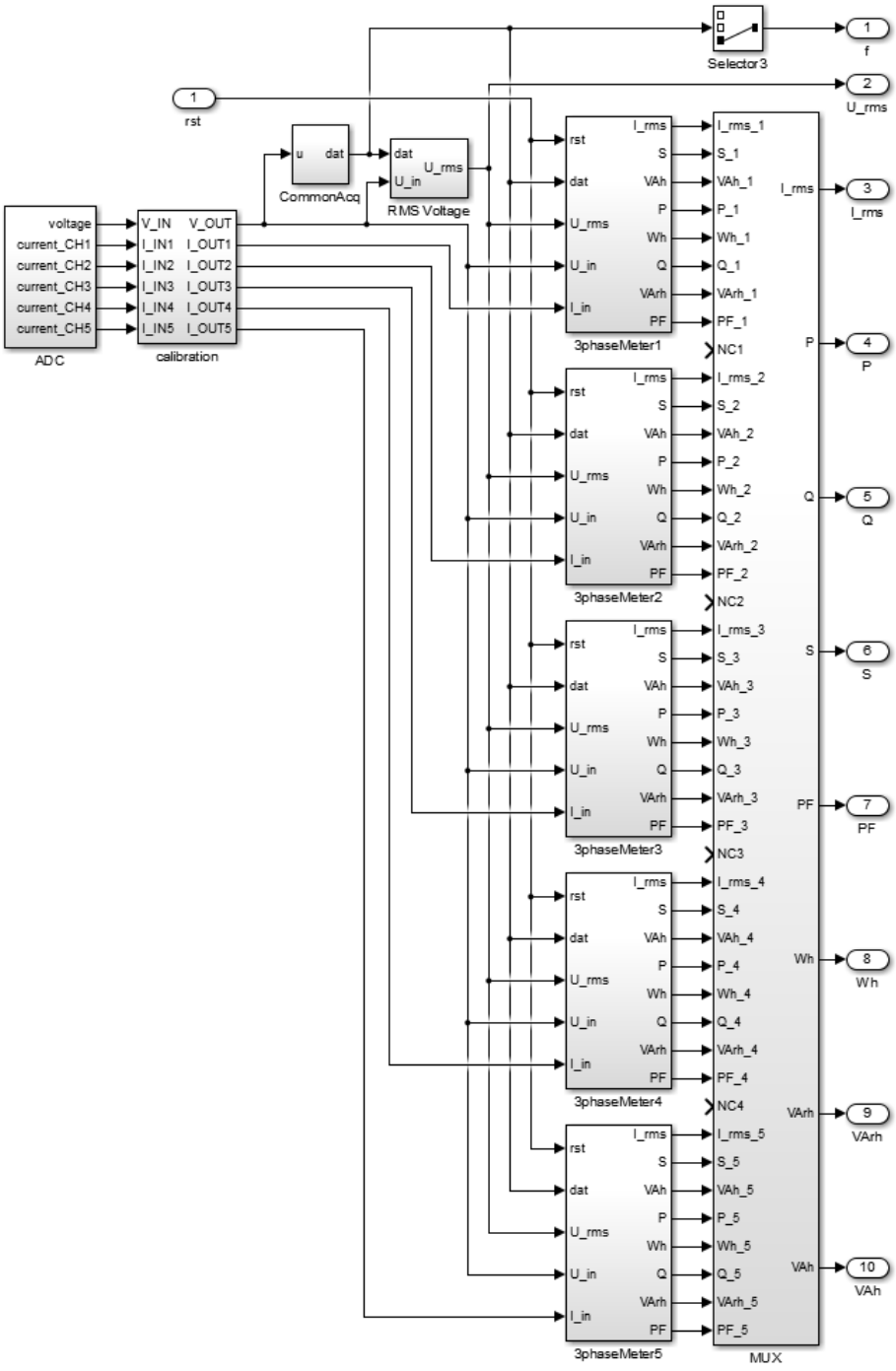


Figure 6.8 *Top layer of acquisition model*

Each of *3phaseMeter* blocks comprises of three *1phaseMeter* subsystems of which one is shown in Figure 6.9. Its main parts are: *Current RMS*, calculating effective value of current, *PhaseDetect* detecting phase shift between voltage and current, *PowerCalculation* and *EnergyCalcualtion*.

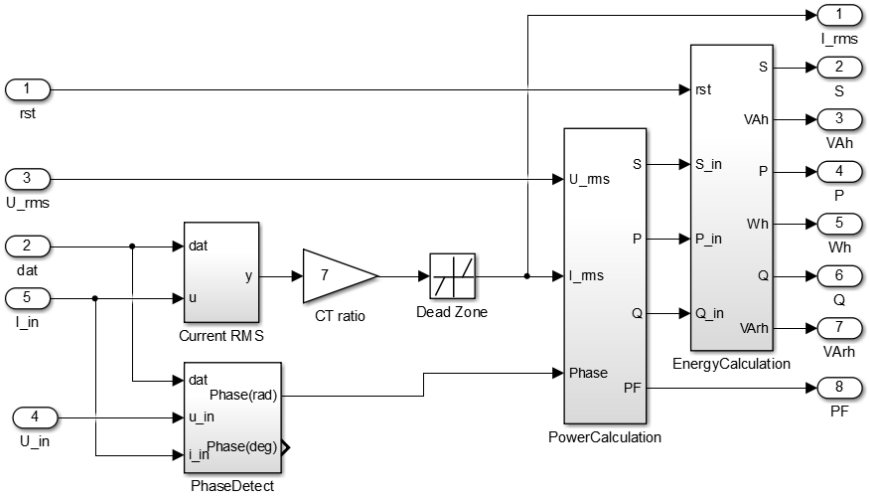


Figure 6.9 *Single-Phase Meter*

An effective value is calculated according (6.4), where u represents the input signal, T is period and U_{RMS} is resulting root-mean-square value.

$$U_{RMS} = \sqrt{\frac{1}{T} \cdot \int_{t}^{t+T} u^2} \tag{6.4}$$

Implementation of this formula into Simulink resulted in model shown in Figure 6.10. Integration is done over two power line cycles (PLCs) that makes period T equal to 40ms. The reason for that was improvement in resolution that is needed because analog input is sampled at relatively slow 2 kHz sampling frequency.

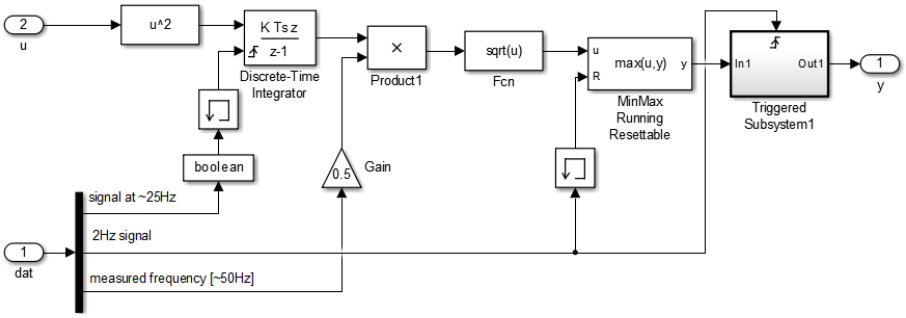


Figure 6.10 *RMS calculation model*

The last two blocks were added to increase stability of readings. It reduces refresh rate of output to 2 Hz that provides much more precise reading as it takes only the maximum reading of 1/2s period.

Phase shift between voltage and current needs to be measured in order to express active and reactive power. It is based on time difference between zero crossing of voltage and current. This difference is divided period that indicates phase shift. A further

processing is required as can be seen from Figure 6.11 below in order to achieve proper output in radians. Low frequency detection has been added to disable displaying wrong values when no load is connected.

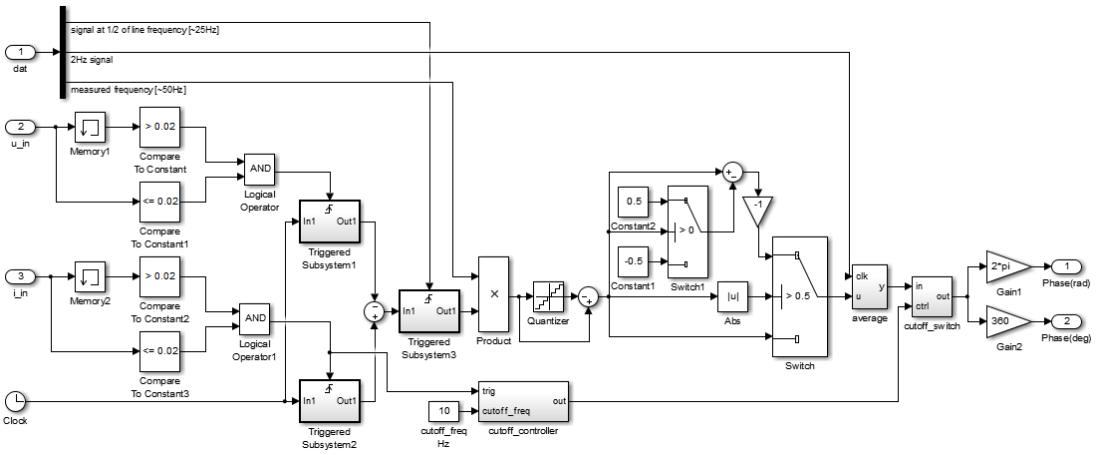


Figure 6.11 *Phase-detect model*

Now, when having effective value of currents and voltages together with a phase angle powers and power factor can be expressed using (6.5).

$$\begin{aligned} S &= U_{RMS} \cdot I_{RMS} \\ P &= U_{RMS} \cdot I_{RMS} \cdot cos(\varphi) \\ Q &= U_{RMS} \cdot I_{RMS} \cdot sin(\varphi) \\ pf &= cos(\varphi) \end{aligned} \tag{6.5}$$

Its implementation is shown in Figure 5.12 together with energy calculators. The energy is calculated from power using discrete integrators. Its outputs had to be divided in order to have per hour energy outputs. Beside that, an external reset has been added to allow reset of these integrators at any time during program execution.

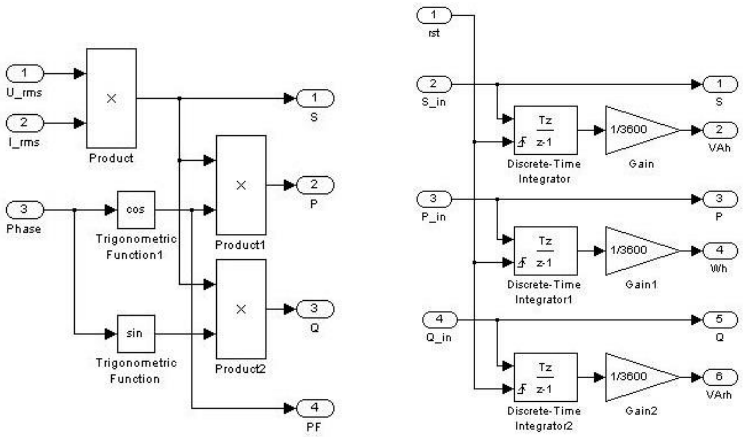


Figure 6.12 *Power and Energy calculation subsystems*

6.4 Data link between dSpace and HEM PC

A data link between dSpace and HEM PC is utilising RS232 interface that is one of two serial interfaces dSpace is equipped with. The maximum transfer speed of this interface is 115200Baud [22]. Baud is the unit for symbol rate or modulation rate in symbols per second or pulses per second [26]. The RS232 is configured to use one start bit and one stop bit and therefore 10bits needs to be transmitted to transmit one byte.

Measurements, generated by algorithm described in chapter 6.3.3, are of double type. This data type uses 64bits that would require transmitting eight bytes for each value. However as the resolution of double type is not utilised a 32bit float type is used instead and therefore only 4bytes needs to be transferred. A data packet comprises of 6 bytes as can be seen on Figure 6.13 below.

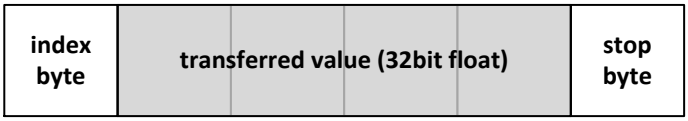


Figure 6.13 *Data packet*

The packet starts with an index byte, followed by four data bytes and terminated with stop byte. The stop byte is of a constant value that is used to synchronize the transmission in case that previous packet was partially lost. The communication should therefore resume without any problems when disconnecting and connecting data cable while the transmission is in progress. The maximum theoretical transmission rate is 1920 packets a second. However, empirically tested reliable speed is about twenty percent slower. Therefore the timer triggering packet transmission is set to generate 1600 pulses a second. If, for example, value of all active powers together with emulated PV production would be sent, an 88 updates a second would be achieved on each of 18 outputs.

A Simulink model on HEM PC is executed in real-time with simulation step size 0.01s that results in receiving block being called hundred times a second. This receiving block is implemented in C and compiled into s-function.

An extra synchronisation feature has been implemented that remotely restarts dSpace when is HEM PC model initialized. This remote reset is done by transmission of one byte with specified delay. This is very convenient as the same behaviour of laboratory setup can be achieved every time.

All parameters of link can be configured through mask that is displayed on Figure 6.14 below.

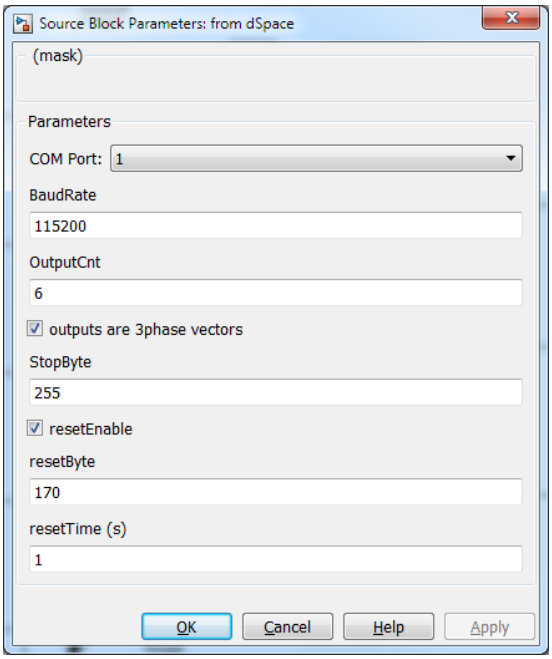


Figure 6.14 *Setting of Digital link on HEM PC*

6.5 HEM PC

A home energy management (HEM) computer is responsible for load control in order to achieve production following and savings on electricity bills. This unit has been implemented using personal computer (PC) with Matlab / Simulink software. This approach eases the development as all data can be easily represented and any modification can be quickly applied. If a required feature is not supported by Matlab then a new block can be programmed thanks to provided application programming interface (API).

HEM PC has to interface with dSpace and with Z-Wave USB transceiver in order to have control over load switching as can be seen from Figure 6.15. This chapter is describing only implementation of Z-Wave interface as interface with dSpace has been covered in chapter 6.4 and the production following algorithm in chapter 4.

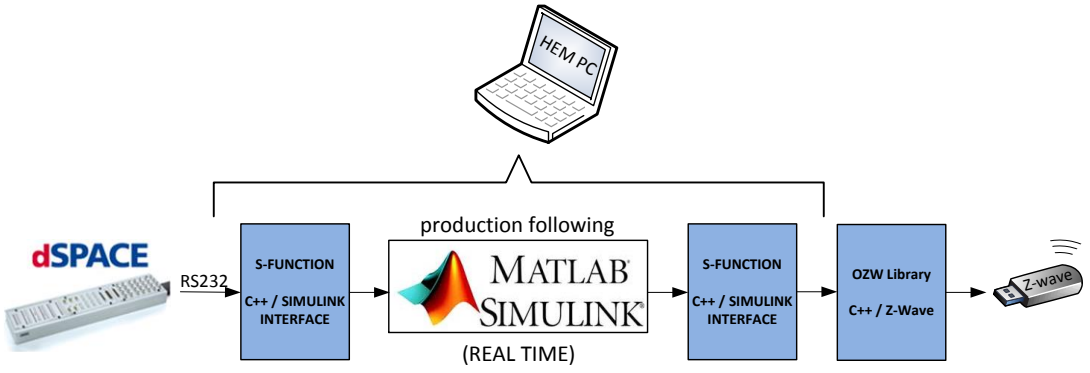


Figure 6.15 *HEM PC and its interface with laboratory equipment*

Z-Wave is wireless technology developed specifically for home automation that provides meshed structure, good range, low-power consumption and secure channel. In order to control Z-Wave relays from Simulink environment a user defined S-Function had to be programmed utilizing Z-Wave communication protocol. Even though Z-Wave protocol is not open, and in order to get access to it one has to buy Z-Wave Alliance membership, there is free alternative. An open-zwave (OZW) project is freely providing source code with documentation on their webpage [27]. This open source code is limited in its functionality, but fortunately this limitation is not affecting use of relay control. Z-Wave radio capability has been added to HEM computer by connecting USB transceiver: Aeon Labs Z-Stick S2. An S-function written in C++ had to be implemented in order to use OZW library from Simulink as shown in Figure 6.15.

First a *zwave_bin_switch.cpp* code has been written that uses API of Simulink and OZW library to make the interface possible. Then a *build.m* script has been written that compiles all OZW library source files to object files that are then linked into final compilation of *zwave_bin_switch.cpp* that results in *zwave_bin_switch.mexw64* file. This MEX file is then loaded into S-Function from Simulink environment. The flow chart of the compilation process is captured in Figure 6.16. The compilation has been done using *mex* function that has been configured to use Microsoft SDKs v7.1 compiler.

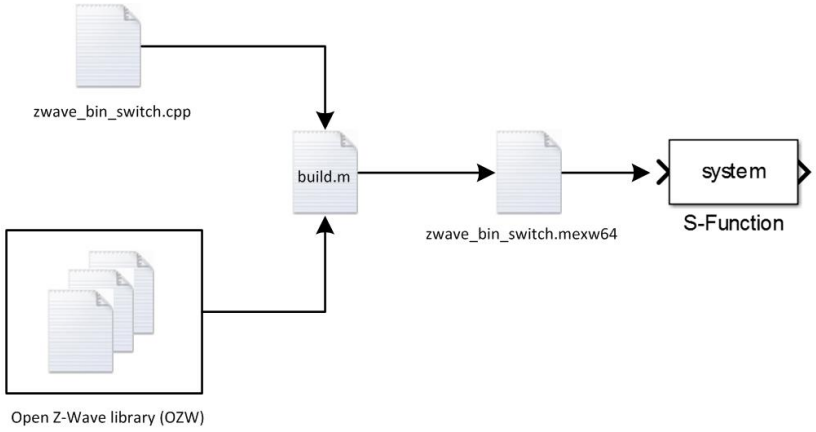


***Figure 6.16** Z-Wave S-Function compilation process*

Compiled MEX function expects four input parameters that can be configure trough mask of S-Function that is shown in Figure 6.17. First parameter is number of serial COM port of Z-Wave radio module. Second parameter is a vector of values corresponding to Z-Wave relay addresses. OZW library generates an XML file that contains identification numbers of all nodes within Z-Wave network that have been previously paired with Z-Wave transceiver. The name of this XML file contains id of Z-Wave USB stick itself. In this case the name of XML is *zwcfg_0x0161e027.xml* that contains id numbers of all relays as shown in Figure 6.18. If there is a problem with XML file generation, then it is recommended to delete all XML files from working directory. It should be noted that first execution during which is the XML file generated can take up to 15minutes.

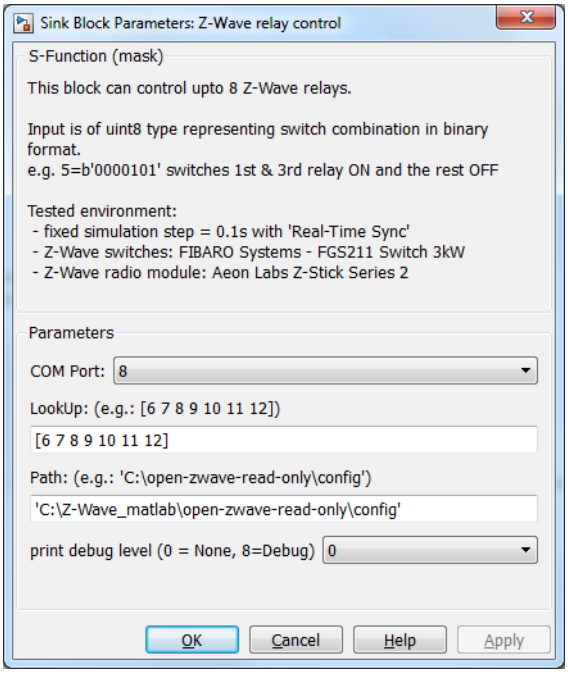


Figure 6.17 *Z-Wave relay control - settings*

```
<?xml version="1.0" encoding="UTF-8"?>
<Driver poll_interval_between="0" poll_interval="500" controller_capabilities="8" api_capabilities="0" node_id="1" home_id="0x0161e027" version="3"
xmlns="http://code.google.com/p/open-zwave/">
  - <Node version="3" query_stage="Complete" max_baud_rate="40000" routing="false" beaming="true" frequentListening="false" listening="true"
    type="Static PC Controller" specific="1" generic="2" basic="2" location="" name="" id="1">
      - <Manufacturer name="Aeon Labs" id="0086">
          <Product type="0002" name="Z-Stick S2" id="0001"/>
        </Manufacturer>
      - <CommandClasses>
          - <CommandClass version="1" name="COMMAND_CLASS_BASIC" id="32" after_mark="true">
              <Instance index="1"/>
              <Value type="byte" index="0" value="0" max="255" min="0" poll_intensity="0" verify_changes="false" write_only="false" read_only="false"
                units="" label="Basic" instance="1" genre="all"/>
            </CommandClass>
        </CommandClasses>
    </Node>
  + <Node version="4" query_stage="Complete" max_baud_rate="40000" routing="true" beaming="true" frequentListening="false" listening="true"
    type="Binary Power Switch" specific="1" generic="16" basic="4" location="" name="" id="6">
  + <Node version="4" query_stage="Complete" max_baud_rate="40000" routing="true" beaming="true" frequentListening="false" listening="true"
    type="Binary Power Switch" specific="1" generic="16" basic="4" location="" name="" id="7">
  + <Node version="4" query_stage="Complete" max_baud_rate="40000" routing="true" beaming="true" frequentListening="false" listening="true"
    type="Binary Power Switch" specific="1" generic="16" basic="4" location="" name="" id="8">
  + <Node version="4" query_stage="Complete" max_baud_rate="40000" routing="true" beaming="true" frequentListening="false" listening="true"
    type="Binary Power Switch" specific="1" generic="16" basic="4" location="" name="" id="9">
  + <Node version="4" query_stage="Complete" max_baud_rate="40000" routing="true" beaming="true" frequentListening="false" listening="true"
    type="Binary Power Switch" specific="1" generic="16" basic="4" location="" name="" id="10">
  + <Node version="4" query_stage="Complete" max_baud_rate="40000" routing="true" beaming="true" frequentListening="false" listening="true"
    type="Binary Power Switch" specific="1" generic="16" basic="4" location="" name="" id="11">
  + <Node version="4" query_stage="Complete" max_baud_rate="40000" routing="true" beaming="true" frequentListening="false" listening="true"
    type="Binary Power Switch" specific="1" generic="16" basic="4" location="" name="" id="12">
</Driver>
```

Figure 6.18 *XML file listing paired devices*

Third parameter is path to configuration files of Z-Wave devices. This folder is part of OZW library that is accessed only when new devices are being added to the network. Last parameter is enabling debugging messages that are printed into Matlab terminal. Setting 0 disables all messages, 6 prints informative messages that are suitable for supervision over behavior of S-Function and 8 informs about all internal processes of Z-Wave protocol.

The Z-Wave S-Function has only one input as the information is represented in binary format. That is reason why the input is of 8bit unsigned integer type. A demon-

stration model shown in Figure 6.19 has been implemented to provide tool for Z-Wave testing.

A manual switches have been used that can be operated while is the model being executed. The output of each switch can be 0 or 1 that is then weighted by binary coefficient representing its position. This coefficient can be expressed as 2^n where n is the position of the switch. If we assume that first position is zero and we have a total of seven positions then the weight go from 1 to 64 as shown in Figure below.

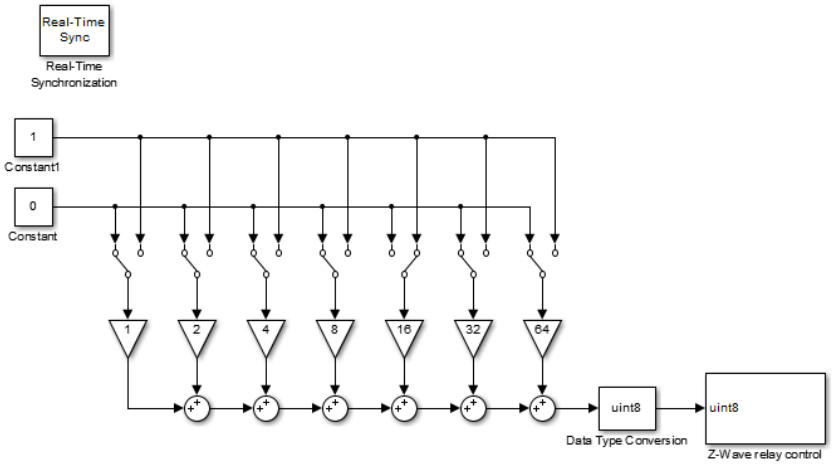


Figure 6.19 *Simulink model for Z-Wave demonstration*

6.6 Laboratory demonstration

A laboratory demonstration is described in this chapter to give basic overview over laboratory operation. This simple setup comprises of Microgrid panel, two 1,5kW loads, dSpace and HEM PC. A photo of this setup is shown in Figure 6.20 below.

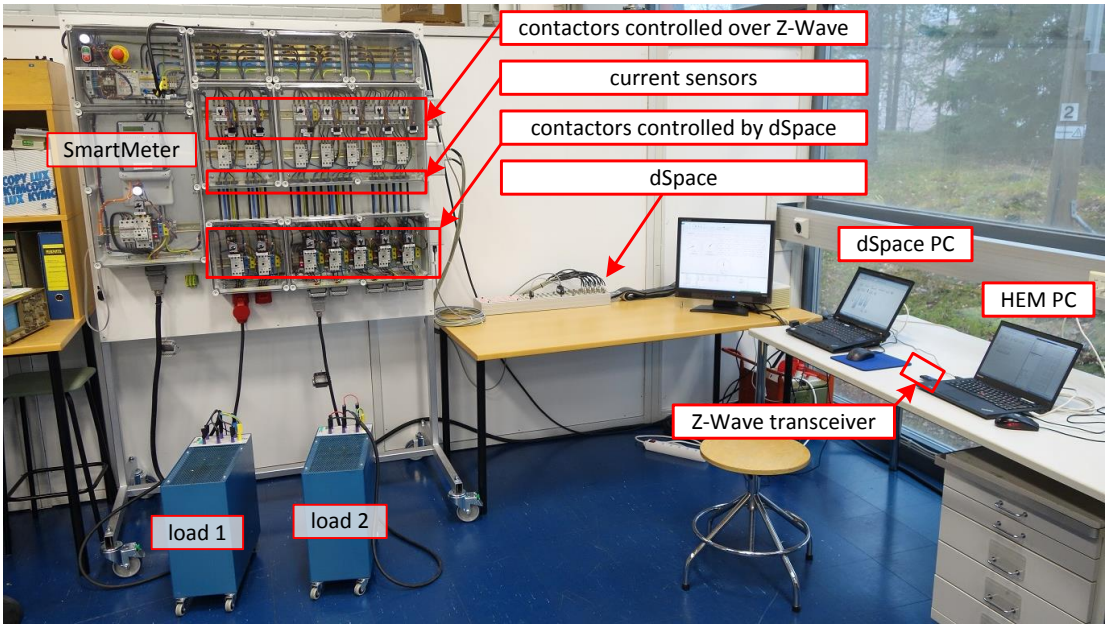


Figure 6.20 *AC Microgrid laboratory*

The dSpace computer, placed on the left-hand side, is displaying real-time measurements as shown in Figure 6.21. The measurements that are displayed are frequency, phase voltages and active powers as they are most relevant for this demonstration. Beside these a controlling signals of contactors are displayed. They are represented by grey color for open state and green color for closed state. A load might not be switched eventhough a contactor is closed. It happens when the second contactor, controlled by Z-Wave, is open. This is the situation of 1st channel in given example depicted in Figures 6.21 and 6.22.

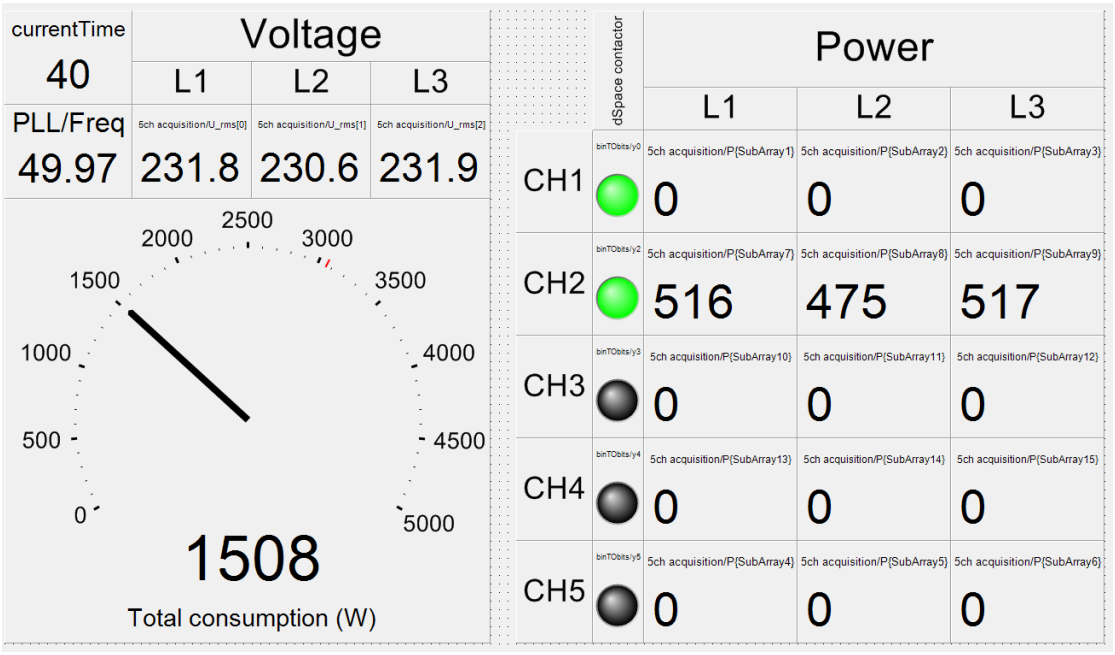


Figure 6.21 *ControlDesk – screen of dSpace PC*

The HEM computer, placed on the right-hand side, is executing simple Simulink model that is shown in Figure 6.22. It has only interface blocks, one with manual control over Z-Wave controlled contactors and second that displays measurements coming from dSpace. It can be seen that the first manual switch is sourced with zero. This is the reason why a load in 1st channel was not switched as discussed above.

The dSpace is providing HEMS PC with readings of power generated by PV panels. These readings originate in data file as the laboratory is not equipped with any PV system. The reason why the data file is interpreted in dSpace and not in HEM PC is the planned future setup in which the dSpace will control PV emulator. More detailed description about planned extensions is introduced in chapter 7.2.

A precision of designed acquisition system has been measured for V_{RMS} and I_{RMS}. A Fluke 79III hand-held multimeter has been used as a reference meter for voltage readings and a current probe Elditest CP6220 together with the same Fluke multimeter has been used as a reference meter for current.

The voltage has been measured at 230VAC at which the error stayed below ±0.5%. The current has been measured at 2A and 9A. The error didn't exceed ±4%.

That is not bad result considering that the accuracy of the current probe itself is specified to be ±3%. The required precision of designed acquisition system is not of great importance and thus calibrating the system using current probe is sufficient. In case that a better accuracy would be required in the future a new calibration could be done using better reference meter. Another way of accuracy improvement is increasing sampling frequency of dSpace. A new dSpace capable of at least 10x faster sampling is planned as one of the future extensions as discussed in chapter 7.2.

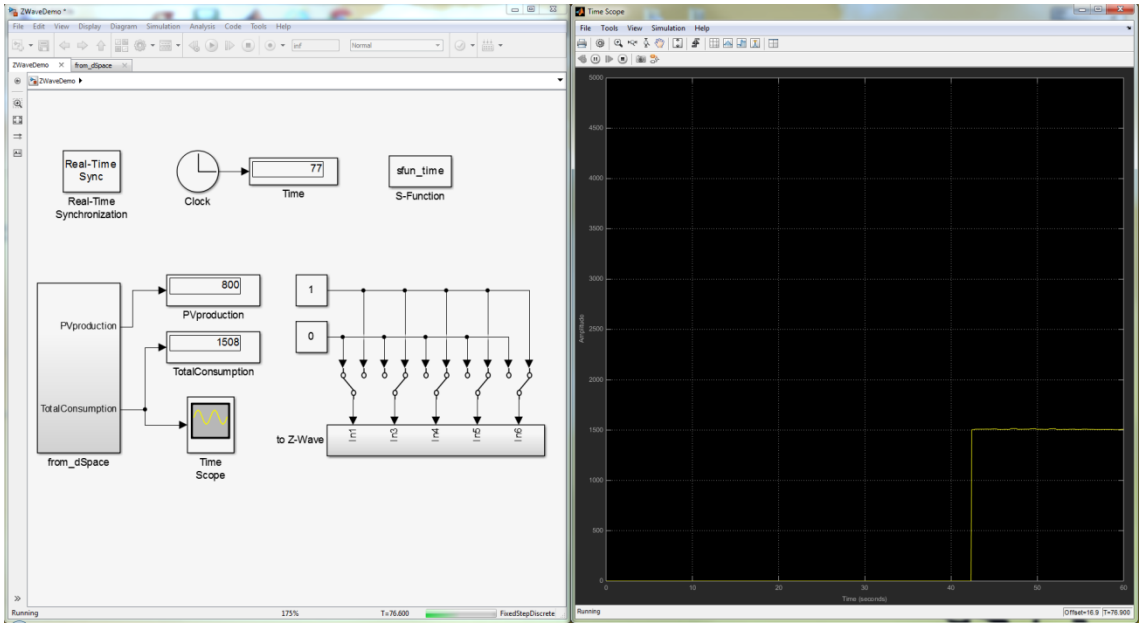


Figure 6.22 *Simulink – screen of HEM PC*

7. DISCUSSION

This chapter is divided into two sections. First, an algorithm, its simulations and ideas for improvements are discussed in chapter 7.1. Then, a future development of AC Microgrid laboratory is introduced in chapter 7.2.

7.1 Algorithm

Case study simulations presented in chapter 5 show that up to 5,6% can be saved on electricity bills. This might not be motivating enough for end customer as the return of investment is around 7 years. Algorithm's implementation is, however, beneficial for network stability as the disturbances, generated by intermittent PV generation, need to be compensated. The system operator has therefore interest in production following implementation so new tariffs can be expected that would increase savings and thus interest of customers.

The main issue with production following algorithm is need of irradiance forecast. Phenomena like morning mist make precise forecast of irradiance close to impossible. In case, that the difference between actual and forecasted PV production would be significant, then energy storage wouldn't have space to store excessive production or too much capacity would be reserved and thus load shifting potential would be curtailed. This issue needs to be emphasized as it is not visible from simulations presented in Figure 5.7. It is because customer sells for only 0.24c/kWh less than is the purchase price. This difference is negligible when compared with price variations in hourly pricing.

The negative effect of error in forecasted temperature and irradiance could be minimized by intra-day rescheduling. Presented algorithm executes scheduler once a day preparing schedule for upcoming day. However, a secondary scheduler could be implemented that would be rescheduling e.g. every hour using the most precise forecast available.

The main strength of algorithm is load shifting implementation using heat energy storage. Use of hot water boiler has appealing benefits like low investment cost and lifetime when compared with battery storage system. It has, however, downsides that should not be ignored. The most important one is the nature of the load itself. Hot water boiler is discrete load whereas power electronics within battery storage system can be continuously controlled. This becomes an issue worth considering when multiple households implement this system. It is because synchronous switching of large discrete

loads would significantly increase peak power consumption of such residential area. In order to reduce this peak power consumption a modification of HEM algorithm would be required. A possibility of remote control of heating appliances timing would be added. Then a superior system responsible for peak power minimization would be added to distribution network.

A simple monetary analysis has been carried out that gives rough estimate of potential savings. Simulation results, presented in Table 5.1, indicate that the return of investment is around 7 years with lifetime savings about 841€.

The real word performance could differ for many reasons. The most important factor is use simplified thermal model of the house. The model doesn't represent open door / window, different insulation materials or heat generated by residents or appliances like oven or PC. Implementation of more realistic model would definitely benefit to this work and should be done in the future.

The HEM computer should be built on cheap, low-power and flexible platform. The presented algorithm uses very little memory and therefore an implementation on cheap ARM-M system-on-the-chip (SoC) processor is possible. It could be e.g. STM32F407 that can be purchased for ~10€. A sensible alternative would be platform based on ARM-A processor that offers significantly more memory and processing power. This platform could be e.g. Raspberry Pi priced at ~35€. The benefit of this platform is that it can run Linux operating system and therefore offer much more flexibility. In case that the algorithm would implement complex optimization method an increased memory requirement would make use of ARM-A processor necessary. It would be therefore reasonable to assume ARM-A based platform when designing new algorithms in the future. Such system would have at least 500MHz processor, 256MB RAM memory and 2GB of storage.

7.2 Future development

The laboratory, presented in chapter 6.6, can be used for basic tests only. However, an extension is planned so more realistic scenarios could be emulated.

The first system to be added to the laboratory is heat energy storage accompanied with radiator in closed-loop system as depicted in Figure 7.1 below. A HEM computer will measure intake and outlet temperature of hot water boiler together with ambient temperature. Charging will be controlled over Z-Wave as described in chapter 6.5 and discharging will be controlled using variable speed water pump. To achieve this control a motor inverter with digital interface has to be used. The water boiler is equipped with thermostat that is not needed as the thermostat is implemented in the software. In order to eliminate operation of embedded thermostat a maximum temperature has to be set. This setup is same as the one used in analysed case-study. It will be therefore interesting to compare results between simulated and emulated setup.

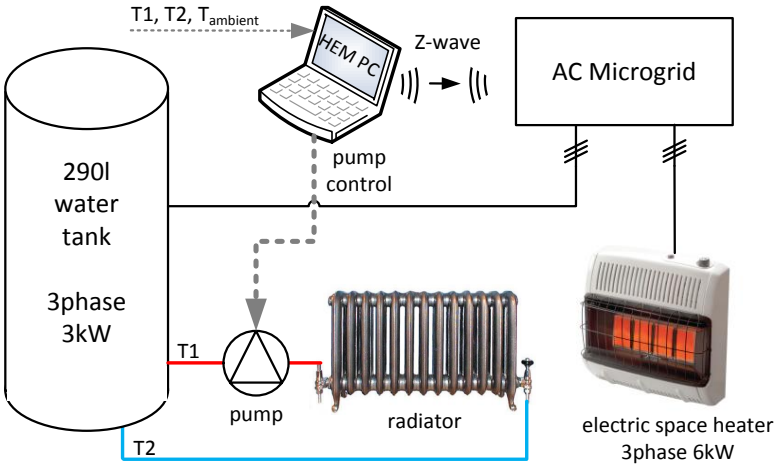


Figure 7.1 *Heat energy storage*

Another system to be added is emulator of photovoltaic production. It will use three-phase grid-tie PV inverter sourced with DC power supply as shown in Figure 7.2 below. A dSpace will control DC source so any irradiance profile can be emulated. The grid-tie inverter will have power control capability so curtailment and power factor control will be possible. It is expected that the power control will bypass MPPT algorithm of grid-tie inverter and thus rectangular I-U profile of DC source should not cause any problem. However, in case that this assumption is wrong, then an output voltage of DC source would be measured by dSpace that would set corresponding current limit so the shape of I-U curve would be as the one of real PV array. A DC source will be 5kW rated unit capable of delivering up-to 500V and 30A manufactured by ElektroAutomatik and the three-phase grid-tie inverter will be rated to 5,8kW provided by ABB.

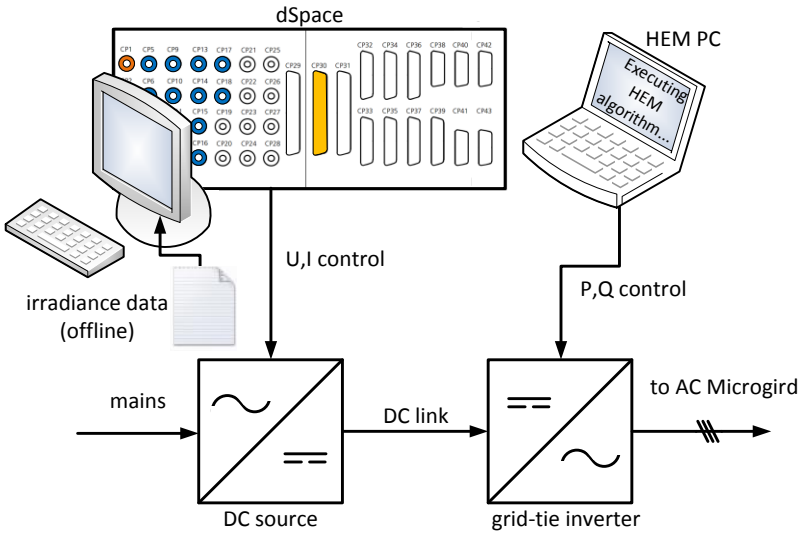


Figure 7.2 *PV production emulator*

Adding battery storage system is the next step of laboratory extension. Its capacity will be ~10kWh operating at ~200V. It will use 90Ah Lithium ion phosphate (LFP) battery

cells from previous project – electric Volkswagen Passat. The exact capacity and voltage will be therefore determined by condition of these cells. This system will utilize new modular battery management system (BMS) designed by Antti Alhonen at TUT. A microcontroller will implement BMS master supervising battery storage operation so all cells will stay within operational range. This BMS master will interface with chargers and HEM PC as shown in Figure 7.3 below.

It is planned to have three Powerfinn PAP3200/160CAN chargers operating in parallel. Choosing three chargers over one bigger charger has been done for two reasons. First, two chargers from previous project could be used and second having three chargers enables active compensation of voltage imbalances in three-phase system. These chargers are 3kW that can provide up to ~54A charging current that translates to charge rate of 0,6C for planned 90Ah, 200V battery pack. The parallel operation will be achieved by controlling one charger in voltage source mode that will provide voltage reference for other two chargers operating in current source mode. These chargers measure output voltage and current that will be read by BMS master as it will manage parallel.

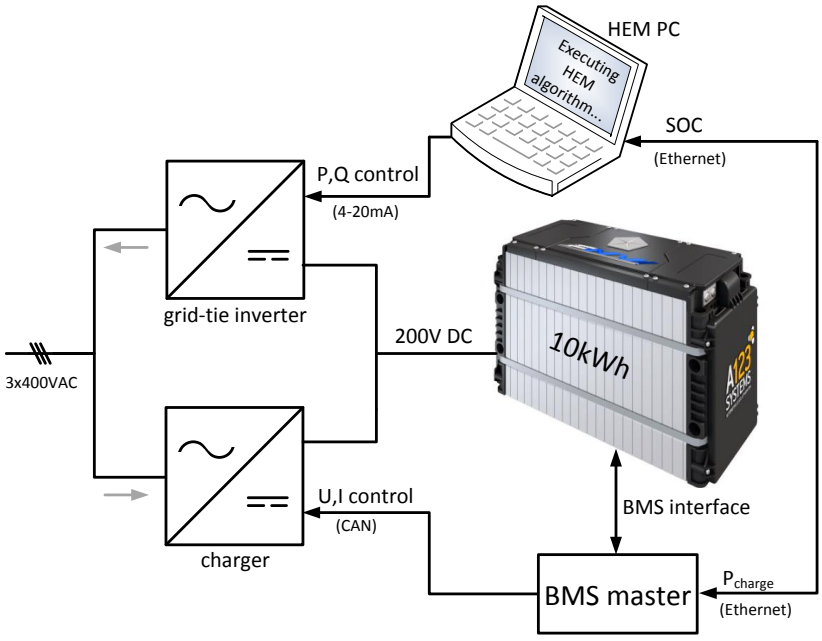


Figure 7.3 *Battery storage system*

Discharging will utilize 10kW, three-phase grid-tie solar inverter from ABB - PVI-10.0-I-OUTD-S. A 10kW model has been chosen as achievable power decreases along with input voltage. Planned battery pack might discharge to ~160VDC at which the maximum power is limited to about 6kW. The power control of the unit can be done two ways. First way is using industrial 4-20mA analog interface that would require USB to 4-20mA conversion. The alternative would be remote control over Ethernet utilising VSN700-5 data logger. It isn't, however, confirmed that application programming interface (API) is provided to develop 3rd party application.

The last extension of laboratory concerns dSpace upgrade. It will get new processor card allowing more than 10x faster sampling resulting in higher precision of measurements. At the same time more analog inputs will be available so all seven three-phase channels could be measured simultaneously. On top of that Ethernet, CAN, RS232, RS422 and RS485 interfaces will be added so future extension will be easier.

8. CONCLUSIONS

Concepts of productions following and load shifting have been implemented in this thesis. Combination of these concepts provides perfect symbiosis as they utilize same equipment.

Case study simulations showed that algorithm's implementation can result in quite significant savings. A simulation of whole year showed that more than 5,6% can be saved on electricity bills. However, this result isn't possible without energy storage system. Simulation results showed that medium size house with total heat transmittance of 100W/K requires about 14kWh storage to achieve this result. To offer reasonable return of investments, cheap energy storage has been used. A hot water boiler with 290litter capacity has been chosen that can store equivalent of 13,5kWh. This setup results in return of investments in less than 7 years whereas expected lifetime is at least twice as long. The estimate of total savings over system's lifetime including all investments is around 841€.

These savings might not seem intriguing enough to make the investment, there are, however, two important factors to consider. First, no support from network operators has been considered, even though they are getting benefits of production following and demand response implementations. It can be therefore expected that a new supportive tariffs would be presented together with participation in investment costs. Secondly, a customer would get tool to manage household's power consumption allowing further savings.

It is important to note that these simulations give only estimate of real-word performance. The achievable savings might differ primarily due to non-ideal thermal model of the house. Therefore, it would be sensible to consider implementation of better model In the future so more trustworthy results could be provided.

All algorithms were implemented in Matlab / Simulink environment to provide maximum flexibility. Interface to AC Microgrid laboratory has been developed so presented algorithms could be easily used in HIL simulations. One of the biggest achievements was successful implementation of Z-Wave wireless interface that enables real-time control from Simulink to wireless relays controlling loads. Moreover a data acquisition system has been developed that allows continuous measurement of all seven three-phase channels of AC Microgrid.

The laboratory is planned to be extended by heat energy storage, battery storage system and emulator of photovoltaic production. Having possibility to fully emulate households' environment will not only confirm simulated results but it will also give more credibility to generated results.

BIBLIOGRAPHY

[1] European Commision, "European statistics," September 2014. [Online]. Available: http://ec.europa.eu/eurostat.

[2] B.K. Bose, "Global warming: energy, environmental pollution, and the impact of power electronics," *IEEE Industrial Electronics Magazine,* pp. 6-17, 2010.

[3] Cha, Hee-Jun, Jin-Young Choi, and Dong-Jun Won, "Smart load management in demand response using microgrid EMS," *Energy Conference (ENERGYCON),* 2014.

[4] Dang, Thanh, and Kathryn Ringland, "Optimal load scheduling for residential renewable energy integration," *Smart Grid Communications (SmartGridComm),* 2012.

[5] Koto, A., Lu, S., Valavaara, T., Rautiainen, A., & Repo, S, "Aggregation of small-scale active resources for smart grid management," *Innovative Smart Grid Technologies (ISGT Europe),* 2011.

[6] Kirschen, Daniel S., "Demand-side view of electricity markets," *IEEE Transactions on 18.2,* pp. 520-527, 2003.

[7] Ideal Grid For All, "Distribution Automation Concept," [Online]. Available: http://webhotel2.tut.fi/units/set/ide4l/D3.1_Final.pdf. [Accessed October 2014].

[8] Alizadeh, Mahnoosh, Anna Scaglione, and Robert J. Thomas., "From packet to power switching: Digital direct load scheduling.," *IEEE Journal on 30.6,* pp. 1027-1036, 2012.

[9] Pipattanasomporn, Manisa, Murat Kuzlu, and Saifur Rahman, "An algorithm for intelligent home energy management and demand response analysis," *IEEE Transactions on 3.4,* pp. 2166-2173, 2012.

[10] Pedrasa, Michael Angelo A., Ted D. Spooner, and Iain F. MacGill., "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services.," *Smart Grid, IEEE Transactions on 1.2,* pp. 134-143, 2010.

[11] Ari Laitinen, Maija Ruska, Göran Koreneff, "Impacts of large penetration of heat pumps on the electricity use," pp. 50-51.

[12] Hens, Hugo S.L.C., "Applied Building Physics : Boundary Conditions, Building Performance and Material Properties," John Wiley & Sons, 2012, p. 238.

[13] Serth, Robert W., Process Heat Transfer: Principles and Applications, Kindlington, GBR: Academic Press, 2007.

[14] Theodore, Louis, "Essential Engineering Calculations Series : Heat Transfer Applications for the Practicing Engineer," John Wiley & Sons, 2011, pp. 257-280.

[15] Bavarian Center for Applied Energy Research, "TRANSYS - Models for Radiator Heating Systems," [Online]. Available: http://www.transsolar.com. [Accessed September 2014].

[16] Andersson, Dan, et al., "Intelligent load shedding to counteract power system instability.," *Transmission and Distribution Conference and Exposition: Latin America, 2004 IEEE/PES. IEEE,* 2004.

[17] Li, Huawei, Yu Fan, and Tao Wu., "Impact of Load Characteristics and Low-Voltage Load Shedding Schedule on Dynamic Voltage Stability.," *Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on. IEEE,* 2006.

[18] TUT/DEE - Jussi Ahola, "DEE Photovoltaic Power Plant Weather Station Data," [Online]. Available: http://www.tut.fi/solar/. [Accessed September 2014].

[19] Weather Underground, September 2014. [Online]. Available: http://www.wunderground.com/weather-forecast/zmw:00000.1.02944.

[20] NordPool, "Nordpool spot," September 2014. [Online]. Available: http://www.nordpoolspot.com/Market-data1/Elspot/Area-Prices/ALL1/Hourly/.

[21] Fortum, "Fortum Lähisähkösopimuksen hinnat," September 2014. [Online]. Available: http://www.fortum.com/countries/fi/yksityisasiakkaat/energiansaasto/sahkon-pientuotanto/hinnastot/pages/default.aspx.

[22] dSpace GmbH, *"DS1103 PPC Controller Board - Features",* 4.1 ed., 2004.

[23] dSpace GmbH, *"DS1103 PPC Controller Board - Hardware Reference",* 4.1 ed., 2004.

[24] Panasonic, "AQ-Q Relays datasheet," [Online]. Available: http://www.panasonic-electric-works.com. [Accessed September 2014].

[25] K. H. Sueker, "Power Electronics Design: A Practitioner's Guide," 2005, pp. 97-100.

[26] C. Murthy, "Data Communication and Networking," Himalaya Publishing House, 2010, p. 102.

[27] openzwave, "open Z-Wave," [Online]. Available: http://www.openzwave.com/. [Accessed September 2014].

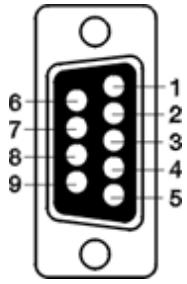
[28] D. S. Lemons, Mere Thermodynamics, Baltimore, MD, USA: Johns Kopkins University Press, 2008.

APPENDIX 1: PROTOTYPE OF AC MICROGRID LABORATORY

An interface between dSpace and solid state relays (SSRs) is implemented in box captured in Figure A.1 below.

Figure A.1 *Photo of SSR control interface box*

Its interface uses CANON DB9 connector described in figure A.2. First seven pins are used to drive solid state relays, eight pin is not connected (N/C) and last ninth pin is used for ground (GND) connection.



1. 1st SSR
2. 2nd SSR
3. 3rd SSR
4. 4th SSR
5. 5th SSR
6. 6th SSR
7. 7th SSR
8. N/C
9. GND

Figure A.2 *pin-out of SSR control interface connector*

Current transducers are integrated inside enclosure of AC Microgrid enclosure right below first row of contactors as shown photo of prototype in Figure A.3. These signals are accessible via the three-phase connectors described in Figure A.5.

Figure A.3 *Photo of three-phase current transducers*

A Figure A.4 shows inside of three-phase voltage transducer. A 6 pin connector on its side can be seen, that provides attenuated signals of all three phases. Its pinout is described in Figure A.5.

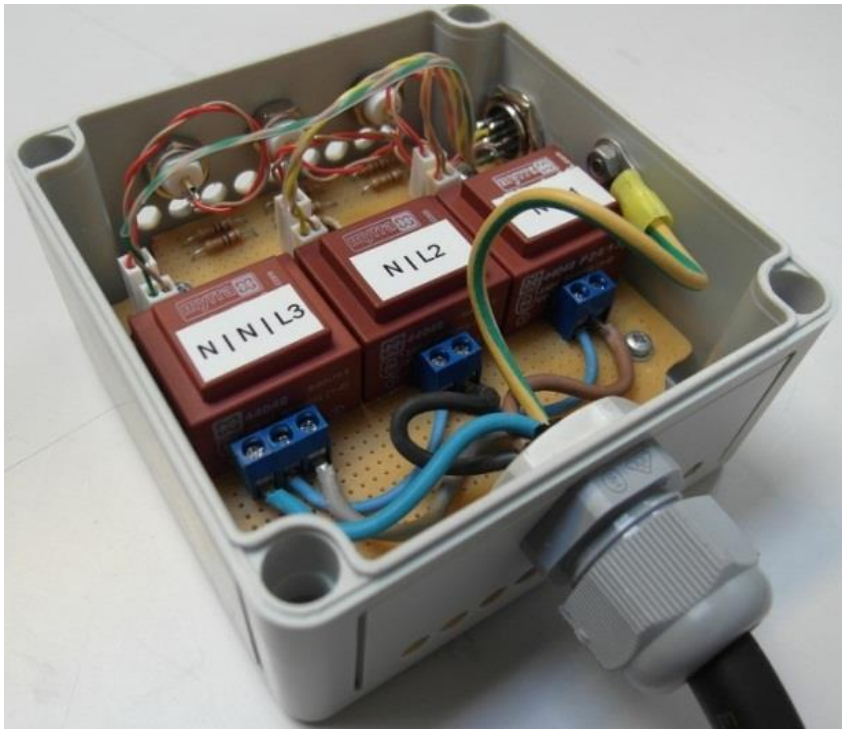
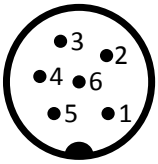


Figure A.4 *Photo of three-phase voltage transducer*



1. L1	orange
2. L1_GND	brown
3. L2	yellow
4. L2_GND	brown
5. L3	green
6. L3_GND	brown

Figure A.5 *pin-out of transducer's three-phase output connector*

The AC Microgrid laboratory with all its accessories is captured in Figure A.6 below.

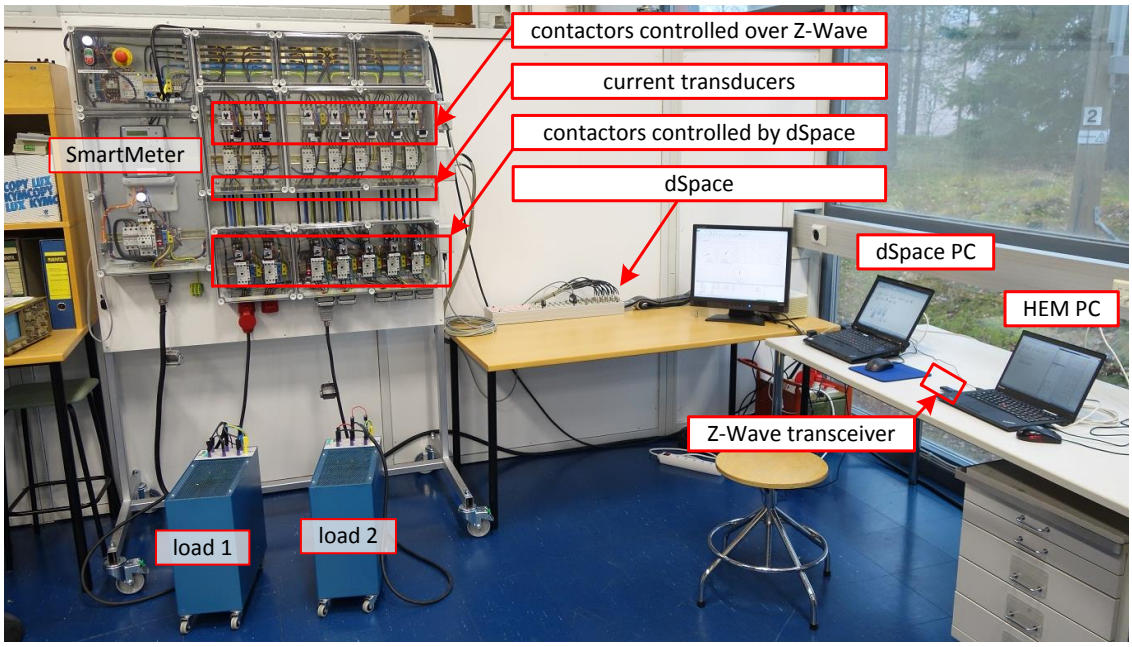


Figure A.6 *AC Microgrid laboratory*

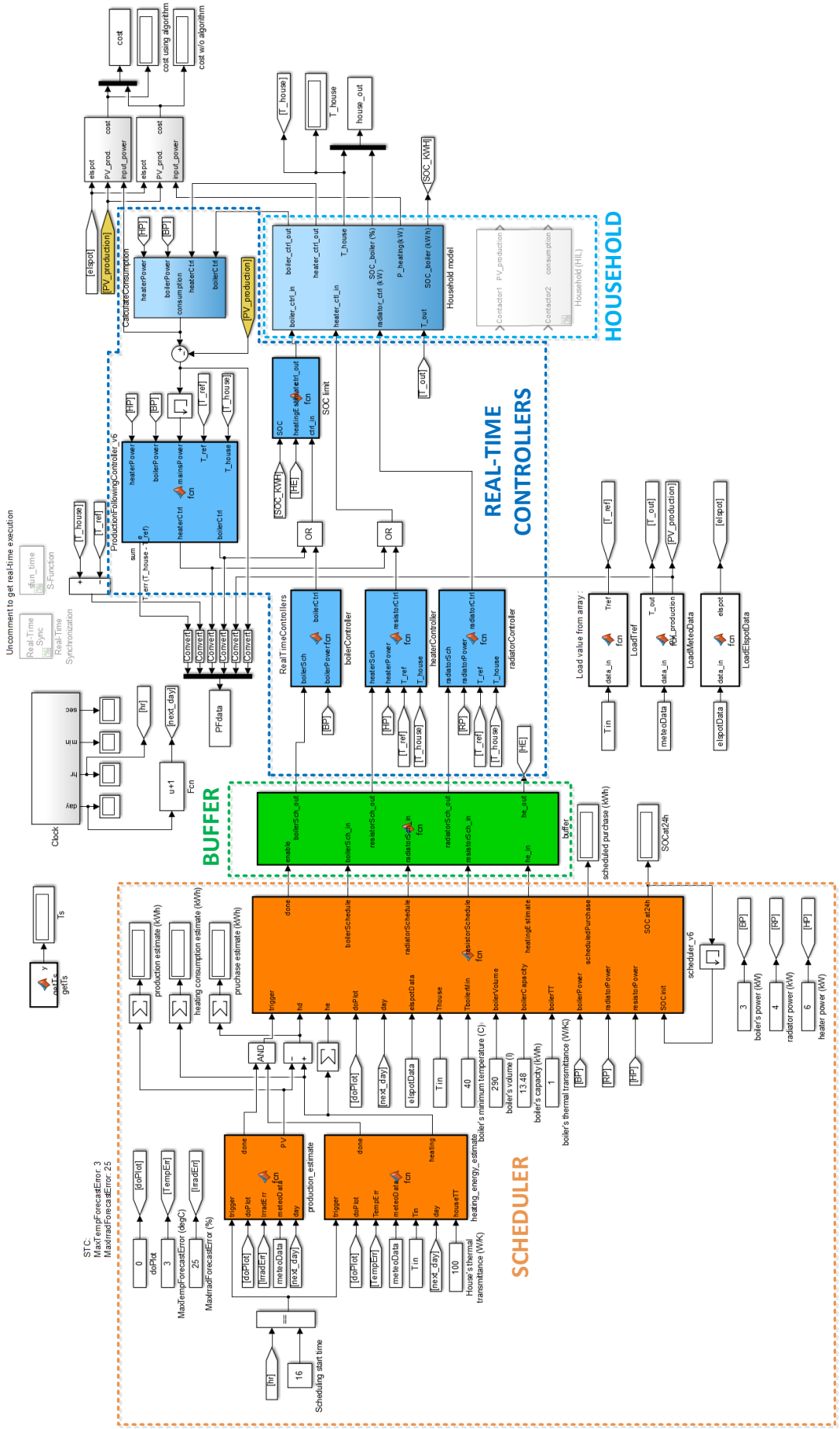


Figure A.7 *Algorithm implementation*